Maksimalisasi Constraint pada Sistem Manajemen Basisdata Relasional

Nahrun Hartono[1)], Erfina[2)]

[1,2]Universitas Islam Negeri Alauddin Makassar
[1,2]Jl. H. M. Yasin Limpo No. 36 Samata, Kab. Gowa, Sulawesi Selatan 0411-841879

E-mail: nahrunhartono@gmail.com[1)], erfina.hisani@uin-alauddin.ac.id[2)]

Abstrak – Basisdata adalah komponen penting dari sebuah aplikasi, basisdata menjadi wadah untuk menyimpan data transaksi pengguna dengan sistem, yang berarti basisdata merupakan sumber informasi. Basisdata yang baik tentu akan mempengaruhi sistem, terutama dalam menyajikan informasi bagi pengguna. Sistem informasi yang baik adalah sistem informasi yang memberikan informasi yang berkualitas. Informasi yang berkualitas tentu ditunjang dengan data data yang berkualitas. Data data yang berkualitas dihasilkan dari struktur basisdata yang berkualitas. Basisdata dapat dirancang sedemikian rupa agar memenuhi kebutuhan pengguna, termasuk dalam hal penginputan data. Pembatasan penginputan data bisa saja dilakukan pada level bahasa pemrograman, namun dalam penelitian ini yang dilakukan adalah melakukan pembatasan penginputan pada level basisdata. *Constraint* merupakan aturan yang ditetapkan saat mendesain basisdata, sehingga pengguna akan dipaksa memasukkan data yang benar yang sesuai dengan aturan bisnis yang ditetapkan. Pada penelitian ini yang dilakukan adalah membuat sekenario pembatasan penginputan data dengan menetapkan aturan data pada setiap kolom. Setelah itu dilakukan penambahan data yang tidak sesuai dengan aturan data untuk mengetahui apakah *constraint* pada kolom tersebut telah sesuai dengan aturan data. Pada penelitian ini digunakan 2 tabel yaitu tabel mahasiswa yang memiliki 6 kolom dan tabel matakuliah yang memiliki 3 kolom, setiap kolom menggunakan *constraint* yang mengikuti aturan data. Hasil pengujian dari masing masing kolom menunjukkan bahwa setiap *constraint* yang digunakan telah sesuai dengan aturan data yang telah ditetapkan.

Kata Kunci: *constraint*, *basisdata*, *aturan data*

Abstract *– Database is an important component of an application. It is a domain for storing data of transaction between users and systems. It is a source of information. A good database definitely will affect the system, especially in providing information to users. A good information system provides qualified information, which clearly supported by qualified data. Qualified data are generated from qualified database structure. Database is designed to meet the users need, including data input. Data input restriction could be defined at the level of programming language. But in this study, it is conducted at the database level. A constraint is a rule determined when designing a database, so that users are forced to input the correct data according to business regulations. In this study, a data input constraint scenario is defined based on data rules for every column. Then, incorrect data are added to observe whether the constraint in the column is properly set or not. Two tables are used, they are table "mahasiswa" and "matakuliah", which contains 6 and 3 columns, respectively. Every column uses the defined constraints. The test result of each column shows that every constraint used has been consistent with the defined data rules.*

Keywords: *constraint, database, data rules*

PENDAHULUAN

Basisdata merupakan sekumpulan data yang akan dikelola sedemikian rupa agar mampu memberikan informasi yang valid kepada penggunannya, baik itu *administrator user* maupun *end user*. Basisdata yang baik tentu akan mempengaruhi sistem informasi, seperti yang dijelaskan dalam prinsip *triad factor* yang mengemukakan performa sistem informasi dapat diukur melalui tiga hal yaitu: 1) desain dan implementasi basisdata, 2) desain dan implementasi aplikasi, 3) *procedure administrative* (*Rob, P. And C Oronel , C. 2009*). Dari ketiga faktor tersebut faktor utama dan pertama yang menjadi tolak ukur performa sistem informasi adalah desain dan implementasi basisdata. Basisdata menjadi sumber informasi bagi sistem informasi, basisdata merupakan wadah yang diatur sedemikian rupa sehingga dapat menampung data data yang dibutuhkan. Sistem informasi yang

baik adalah sistem informasi yang memberikan informasi yang berkualitas. Informasi yang berkualitas tentu ditunjang dengan data data yang berkualitas. Data data yang berkualitas dihasilkan dari struktur basisdata yang berkualitas. (Raharjo, 2012)

Secara umum sistem basisdata relasional terbagi menjadi tiga komponen utama yaitu: 1) Data, yang merupakan informasi yang disimpan dalam struktur yang terintegrasi. Umumnya data data tersebut disimpan ke dalam kolom kolom pada basis data. 2) *Hardware*, yang merupakan perangkat keras berupa *Personal Computer* (PC) dan media penyimpanan. 3) *Software*, yang merupakan perangkat lunak yang digunakan untuk mengelola basis data. Ada banyak pilihan *software* basis diantaranya adalah MySQL, PostgreSQL, DB2, SQL Server, Orcale, dan lain lain. 4) user, merupakan pengguna yang berupa seseorang yang mengelola basisdata (*basisdata administrator*) ataupun yang mengambil hasil pengelolaan basisdata melalui bahasa query (*end user*) yang menggunakan data yang tersimpan dan terkelola (Utami & Raharjo, 2006). Tabel, kolom dan record merupakan element penting dari basis data. Dalam satu basis data mungkin saja terdapat banyak tabel, dalam tabel tabel tersebut mungkin saja memiliki banyak kolom yang didalamnya berisi *record* atau data (Hartono et al., 2016). Pengelolaan basis data menggunakan bahasa tersendiri yaitu bahasa SQL (*Struktur Query Language*) yang umumnya disebut dengan query. Bahasa SQL dikelompokkan menjadi tiga kategori yaitu 1) *Data Definition Language* (DDL), yang digunakan mendefinisikan struktur basis data. Pembuatan tabel, pembuatan kolom dan *constraint* didefinisikan menggunakan perintah *Data Definition Language*. 2) *Data Manipulation Language*, yang digunakan untuk memanipulasi dan mengakses data (Darmanto, 2015).

Jaminan kualitas data pada basisdata dapat dilakukan dengan berbagai cara salah satu diantaranya adalah dengan memberikan batasan inputan data yang akan dimasukkan kedalam basisdata. Pembatasan inputan data dapat dilakukan dengan menggunakan *constraint*. *Constraint* merupakan aturan yang ditetapkan saat mendesain basisdata, sehingga *user* akan dipaksa memasukkan data yang benar yang sesuai dengan aturan bisnis yang ditetapkan (Raharjo & Istiyanto, 2003). Secara umum *constraint* dikelompokkan ke dalam tiga tipe *constraint*, yaitu: 1) *Entity Constraint*, 2) *Referensial Constraint*, 3) *Domain Constraint*. *Constraint* umumnya digunakan untuk membatasi inputan, *constraint* merupakan struktur pada basisdata yang dibuat oleh perancang basisdata untuk mencerminkan tingkah laku tabel berdasarkan fakta dan realitas yang ada (Raharjo, 2013). Umumnya saat mendesain basisdata hanya menggunakan tipe data yang tergolong dalam *domain constraint* serta *primary key* yang tergolong dalam *entity constraint*. *Primary key* merupakan atribut yang digunakan sebagai identifikasi untuk membedakan satu baris data dengan baris lainnya dalam satu tabel sedangkan tipe data digunakan untuk memastikan data yang akan ditambahkan ke dalam kolom pada tabel sesuai dengan tipe data yang telah ditentukan. Penggunaan *domain constraint* dengan menggunakan tipe data dan *entity constraint* dengan menggunakan *primary key* tidak memberikan jaminan terhadap terhadap kesesuaian data dengan fakta yang ada. Sebagai contoh dalam satu tabel yang berisi data penduduk, tentunya akan memiliki kolom nama dengan tipe data *variable character* atau *character*. Berdasarkan tipe data maka ada kemungkinan data yang ditambahkan pada kolom nama tersebut dapat berupa simbol, angka, huruf atau gabungan dari ketiganya sementara faktanya tidak ada nama yang menggunakan karakter simbol atau gabungan antara huruf dan simbol. Memaksimalkan penggunaan *constraint* akan mengurangi ketidaksesuaian data dengan fakta yang sebenarnya. Penelitian ini akan memberikan gamabaran dalam memaksimalkan penggunaan *constraint* pada struktur basisdata.

METODOLOGI PENELITIAN

Penelitian ini merupakan penelitian eksperimen, menurut Arboleda yang disarikan dalam Setyanto, A.E. (2013) pengertian eksperimen adalah metode penelitian dengan cara melakukan manipulasi satu atau lebih variabel dengan sengaja untuk dicari pengaruhnya pada satu atau variabel lain yang diteliti. Variabel itu sendiri terbagi menjadi dua yaitu variabel bebas adalah variabel yang dimanipulasi, sedangkan variabel terikat adalah variabel yang dilihat pengaruhnya (Setyanto, 2013).

Pada penelitian ini yang akan dilakukan adalah membuat skenario aturan data, dimana aturan data tersebut merupakan aturan yang ditetapkan untuk menentukan model data yang harus akan diinputkan kedalam kolom. Aturan data tersebut akan diterapkan

pada kolom di basisdata. Setiap kolom pada basisdata yang dirancang akan mengikuti aturan yang telah ditetapkan dengan menggunakan *constraint*, setelah itu akan dilakukan pengujian dengan menginputkan data pada kolom tersebut untuk mengetahui kesesuaian antara aturan data dengan constraint yang dibuat pada kolom basisdata.

HASIL DAN PEMBAHASAN

Penelitian yang dilakukan adalah dengan menerapkan *Entity Constraint*, *Referensial Constraint*, *Domain Constraint* pada tabel, pada tabel tersebut akan dilakukan uji coba *constraint* yang telah diterapkan. Adapun *constraint* yang akan diterapkan adalah *Primary Key*, *Foreign Key*, *Type Data* dan *Check Constraint*. Setiap *constraint* pada tabel tersebut akan dilakukan uji coba untuk mengetahui *constraint* tersebut bekerja. Adapun aturan data yang akan digunakan adalah aturan data mahasiswa secara umum. Masing masing Perguruan Tinggi umumnya memiliki data mahasiswa dan memiliki aturan data sendiri sebagai contoh Nomor Induk Mahasiswa (NIM) yang terdiri dari 10 karakter angka. Adapun aturan data yang diterapkan pada masing masing kolom ditunjukkan tabel 1 berikut:

Tabel 1. Skenario aturan data

Data	Kolom	Aturan Data
Mahasiswa	nim	▪ Tidak kurang atau lebih dari 11 karakter angka ▪ Hanya bernilai karakter angka ▪ Tidak bernilai NULL
	nama	▪ Nama maksimal 40 karakter ▪ Nama bernilai karakter huruf, angka dan spasi ▪ Nama tidak menggunakan karakter simbol ▪ Nama tidak bernilai NULL
	tempat_lahir	▪ Tempat lahir maksimal 35 karakter] ▪ Hanya menggunakan karkater huruf, angka dan spasi ▪ Diperbolehkan menggunakan karakter simbol garis mendatar (-), selain dari karakter tersebut tidak diperbolehkan
		▪ Tidak bernilai NULL
	tanggal_lahir	▪ Tidak bernilai NULL ▪ Umur harus diatas 18 tahun
	gender	▪ Kolom gender hanya bernilai 1 karakter huruf, L yang berarti laki laki dan P yang berarti perempuan ▪ Tidak bernilai NULL
	kode_mk	▪ Data kode_mk sama dengan data yang ada pada kode_mk pada tabel matakuliah
matakuliah	kode_mk	▪ Maksimal 5 karakter ▪ Kode_mk hanya bernilai karakter huruf dan angka ▪ Kode_mk tidak diperbolehkan menggunakan spasi dan karakter simbol ▪ Tidak bernilai NULL
	nama_mk	▪ Nama maksimal 40 karakter ▪ Nama adalah karakter huruf, angka dan spasi serta tidak menggunkan karakter simbol
	sks	▪ Hanya bernilai angka ▪ Tidak diperbolehkan menggunakan huruf atau simbol ▪ Nilai minimal sks adalah 1 (satu) ▪ Nilai maksimal sks adalah 10 (sepuluh)

Pada tabel 1 diatas ditunjukkan ada dua tabel yang digunakan, yaitu tabel mahasiswa dan tabel matakuliah, tabel mahasiswa memiliki 6 kolom dan tabel matakuliah memiliki 3 kolom. Setiap kolom pada kedua tabel tersebut memiliki aturan data tersendiri. Aturan aturan data tersebut merupakan acuan dalam pembuatan tabel dan kolom pada sistem basisdata, adapun basisdata yang telah dibuat pada penelitian ini adalah basisdata akademik dengan struktur tabel matakuliah dan tabel mahasiswa ditunjukan pada gambar 1 dan gambar 2 berikut:

```
akademik=# \d matakuliah
                    Table "public.matakuliah"
 Column  |         Type          | Collation | Nullable | Default
---------+-----------------------+-----------+----------+---------
 kode_mk | character(5)          |           | not null |
 nama_mk | character varying(40) |           | not null |
 sks     | integer               |           | not null | 0
Indexes:
    "matakuliah_pkey" PRIMARY KEY, btree (kode_mk)
Check constraints:
    "check_kode_mk" CHECK (kode_mk ~* '^[a-z,0-9,.,\,,]+$'::text)
    "check_nama_mk" CHECK (nama_mk::text ~* '^[a-z,0-9,.,\,,]+$'::text)
    "check_sks" CHECK (sks > 0 AND sks < 11)
Referenced by:
    TABLE "mahasiswa" CONSTRAINT "mahasiswa_kode_mk_fkey" FOREIGN KEY (kode_mk)
 REFERENCES matakuliah(kode_mk)
```

Gambar 1. Struktur tabel matakuliah

```
akademik=# \d mahasiswa
                    Table "public.mahasiswa"
    Column     |         Type          | Collation | Nullable | Default
---------------+-----------------------+-----------+----------+---------
 nim           | character(11)         |           | not null |
 nama          | character varying(40) |           | not null |
 tempat_lahir  | character varying(35) |           | not null |
 tanggal_lahir | date                  |           | not null |
 gender        | character(1)          |           | not null |
 kode_mk       | character(5)          |           | not null |
Indexes:
    "mahasiswa_pkey" PRIMARY KEY, btree (nim)
Check constraints:
    "check_gender" CHECK (gender = ANY (ARRAY['L'::bpchar, 'P'::bpchar]))
    "check_kode_mk" CHECK (kode_mk ~* '^[a-z,0-9,.,\,,]+$'::text)
    "check_nama" CHECK (nama::text ~* '^[a-z,.,\,, ]+$'::text)
    "check_nim" CHECK (nim ~* '^[0-9+$'::text)
    "check_tanggal_lahir" CHECK (tanggal_lahir < (CURRENT_DATE - '18 years'::interval year))
    "check_tempat_lahir" CHECK (tempat_lahir::text ~* '^[a-z,.,\,, ]+$'::text)
Foreign-key constraints:
    "mahasiswa_kode_mk_fkey" FOREIGN KEY (kode_mk) REFERENCES matakuliah(kode_mk)
```

Gambar 2. Struktur tabel mahasiswa

Gambar 1 dan gambar 2 menunjukkan struktur tabel matakuliah dan struktur tabel mahasiswa, tabel tabel tersebut mengikuti aturan data yang ditunjukkan pada tabel 1. Hal ini dapat dilihat pada penggunaan *constraint* pada kedua struktur tabel tersebut. Namun untuk menguji apakah kolom kolom pada tabel tersebut telah sesuai dengan aturan data yang ditunjukkan pada tabel 1, maka masing masing kolom akan dibuat terpisah dalam suatu tabel. Agar dapat menunjukkan pengujian yang sebenar benarnya maka penamaan tabel akan diikuti dengan simbol underscore (_) yang kemudian diikuti dengan nama kolom yang akan diuji coba, sebagai contoh untuk menguji kolom kode_mk maka akan dibuat tabel matakuliah_kode_mk yang didalam tabel tersebut hanya memliki satu kolom yaitu kolom kode_mk. Selanjutnya akan diinputkan data pada kolom yang telah dibuat dengan menggunakan perintah *INSERT*. Data yang diinputkan adalah data yang tidak sesuai dengan aturan data yang ditetapkan.

Kolom yang pertama kali dilakukan ujicoba adalah kolom kode_mk pada tabel matakuliah, dapat dilihat pada tabel 1 bahwa kolom kode_mk memiliki aturan data yaitu 1) maksimal 5 karakter, 2) kode_mk hanya bernilai karakter huruf dan angka, 3) kode_mk tidak diperbolehkan menggunakan spasi dan karakter simbol, 4) Tidak bernilai NULL, adapun hasil pengujian penginputan data yang tidak sesuai dengan aturan pada gambar 3, gambar 4, gambar 5 dan gambar 6 berikut :

```
akademik=# INSERT INTO matakuliah_kode_mk (kode_mk)
akademik-# VALUES ('KODEMK-1');
ERROR:  value too long for type character(5)
akademik=#
```

Gambar 3. Pengujian 1 kolom kode_mk: maksimal 5 karakter

```
akademik=# INSERT INTO matakuliah_kode_mk (kode_mk)
VALUES ('SBD-1');
ERROR:  new row for relation "matakuliah_kode_mk" violates check constraint "check_kode_mk"
DETAIL:  Failing row contains (SBD-1).
akademik=#
```

Gambar 4. Pengujian 2 kolom kode_mk: hanya bernilai angka dan huruf

```
akademik=# INSERT INTO matakuliah_kode_mk (kode_mk)
VALUES ('');
ERROR:  new row for relation "matakuliah_kode_mk" violates check constraint "check_kode_mk"
DETAIL:  Failing row contains (     ).
```

Gambar 5. Pengujian 3 kolom kode_mk: tidak menggunakan spasi

```
akademik=# INSERT INTO matakuliah_kode_mk (kode_mk)
akademik-# VALUES (NULL);
ERROR:  null value in column "kode_mk" violates not-null constraint
DETAIL:  Failing row contains (null).
```

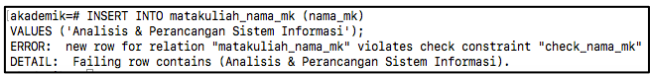
Gambar 6. Pengujian 4 kolom kode_mk: tidak bernilai NULL

Gambar 3 sampai dengan gambar 6 merupakan pengujian kesesuaian aturan data yang diterapkan pada basisdata dengan menggunakan *constraint*. Dapat dilihat bahwa empat aturan data yang telah ditetapkan dapat diterapkan pada basisdata menggunakan *constraint*, gambar 3 menunjukkan pembatasan batas jumlah karakter yang bisa diterima pada kolom kode_mk, gambar 4 menunjukkan pembatasan karakter inputan dimana dapat dilihat data yang diinputkan merupakan gabungan antara angka huruf dan simbol sementara aturan data pada tabel 1 untuk kolom kode_mk disebutkan bahwa kolom kode_mk hanya menerima karakter huruf dan angka. Gambar 5 menujukkan kesalahan penginputan jika tidak ada nilai yang diberikan pada kolom kode_mk seperti yang disebutkan pada tabel 1 bahwa kolom kode_mk tidak menerima spasi atau nilai kosong dan gambar 6 kesalahan penginputan data jika data yang diinputkan bernilai NULL.

Kolom selanjutnya yang diujicoba adalah kolom nama_mk untuk pengujian kolom tersebut seperti yang telah disebutkan sebelumnya, maka akan dibuat tabel dengan nama matakuliah_nama_mk. Hasil pengujian dapat dilihat pada gambar 6 dan gambar 7 berikut :

```
akademik=# INSERT INTO matakuliah_nama_mk (nama_mk)
VALUES ('Analisis dan Perancangan Sistem Informasi');
ERROR:  value too long for type character varying(40)
```

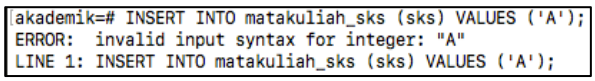
Gambar 7. Pengujian 1 kolom nama_mk: maksimal 40 karakter



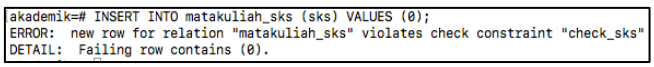
Gambar 8. Pengujian 2 kolom nama_mk: hanya bernilai angka, huruf dan spasi

Gambar 7 dan 8 merupakan pengujian kesesuaian aturan data dengan menggunakan *constraint*, dapat dilihat aturan data pada kolom nama_mk adalah jumlah karakter maksimal 40 karakter pada pengujian yang ditunjukkan pada gambar 7 dapat dilihat secara otomatis sistem basisdata menolak penginputan data karena jumlah karakter yang diinpukan melebihi aturan data. Sama halnya dengan yang terlihat pada gambar 8, gambar tersebut merupakan pengujian untuk menguji aturan data kedua pada kolom nama_mk yaitu: kolom nama_mk hanya bernilai angka, huruf dan spasi. Gambar 8 menunjukkan kesalah penginputan, dimana data yang diinputkan terdapat karakter simbol sehingga secara otomatis sistem basisdata menolak data tersebut.

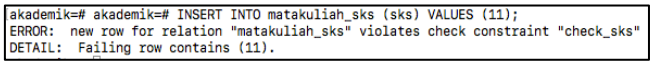
Selanjutnya kolom yang diuji coba adalah kolom sks, untuk itu perlu dibuat sebuah tabel dengan nama makuliah_sks yang didalamya terdapat satu kolom sks. Adapun aturan data pada kolom sks yaitu: 1) hanya bernilai angka, 2) tidak diperbolehkan menggunakan huruf atau simbol, 3) nilai minimal sks adalah 1 (satu), 4) nilai maksimal sks adalah 10 (sepuluh). Gambar 9, gambar 10 dan gambar 11 menunjukkan hasil pengujian pada kolom sks:



Gambar 9. Pengujian 1 kolom nama_mk: hanya bernilai angka



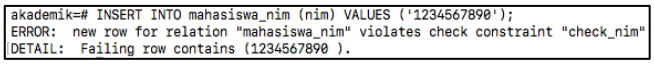
Gambar 10. Pengujian 2 kolom nama_mk: nilai sks minimal 1



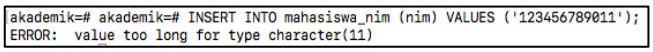
Gambar 11. Pengujian 3 kolom nama_mk: nilai sks maksimal 10

Gambar 9 menunjukkan pengujian aturan data pertama dan aturan data kedua, dapat dilihat pada gambar 9 diinputkan data dengan nilai berupa karakter huruf sehingga ditolak oleh sistem basidata, sementara aturan data menunjukkan bahwa kolom tersebut hanya menerima inputan berupa angka. Pada gambar 10 dan 11 menunujukkan pengujian aturan data 3 dan 4, pada pengujian dilakukan pengiputan data pada kolom sks tabel matakuliah_sks dengan nilai 0 (nol) dan yang berikutnya di inputkan data dengan nilai 11 (sebelas) sementara aturan data mengharuskan data yang diinputkan ke kolom tersebut adalah bernilai minimal 1 dan maksimal 10, sehingga secara otomatis ditolak oleh sistem karena tidak sesuai aturan data yang telah ditetapkan.

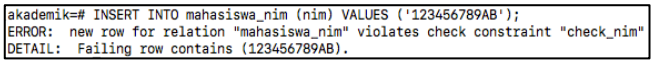
Tahap selanjutnya adalah melakukan pengujan pada tabel mahasiswa yang memiliki 7 (tujuh) kolom. Seperti yang telah dilakukan sebelumnya, setiap kolom pada tabel mahasiswa akan dibuat pada satu tabel. Adapun kolom yang pertama kali dilakukan pengujian adalah kolom nim, adapun aturan data untuk kolom nim adalah 1) Tidak kurang atau lebih dari 11 karakter angka, 2) hanya bernilai karakter angka, 3) Tidak bernilai NULL. Adapun hasil pengujian kolom nim dapat dilihat pada gambar 12, gambar 13, gambar 14 dan gambar 15 berikut:



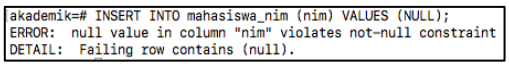
Gambar 12. Pengujian 1.1 kolom nim: tidak kurang dari sebelas karakter



Gambar 13. Pengujian 1.2 kolom nim: tidak lebih dari sebelas karakter



Gambar 14. Pengujian 2 kolom nim: hanya bernilai karakter angka



Gambar 15. Pengujian 3 kolom nim: tidak bernilai NULL

Gambar 12, 13, 14, 15 merupakan hasil pengujian kolom nim, dapat dilihat untuk pengujian pertama (gambar 12) nilai yang diinputkan kurang dari 11 karakter, dimana nilai yang diinputkan berjumlah 10 karakter angka sehingga secara otomatis sistem basisdata menolak penginputan data tersebut. pengujian kedua (gambar 13) nilai yang diinputkan lebih dari 11 karakter dimana dapat dilihat nilai yang diinputkan berjumlah 12 karakter angka sehingga sistem basisdata menolak data yang diinputkan. Pengujian ketiga (gambar 14) dilakukan dengan menambahkan data nilai karakter angka yang digabungkan dengan karakter huruf yang berjumlah

11 karakter, secara otomatis sistem basisdata menolak penginputan tersebut karena menurut aturan data yang telah ditetapkan kolom nim hanya menerima karakter angka. Pengujian keempat (gambar 15) dilakukan dengan menginputkan nilai NULL pada kolom nim dan secara otomatis sistem basisdata menolak nilai inputan tersebut.

Tahap selanjutanya dalah pengujian kolom nama, berdasarkan aturan data yang telah ditetapkan maka diketahui aturan data pada kolom nama adalah 1) nama maksimal 40 karakter, 2) nama bernilai karakter huruf, angka dan spasi, 3) nama tidak menggunakan karakter simbol, 4) nama tidak bernilai NULL, berdasarkan aturan data tersebut maka dibuat kolom nama dengan *constraint* yang disesuaikan dengan aturan aturan data. Untuk pengujian 1 dan pengujian 4 tidak dilakukan pengujian karena tindakan pengujian sama dengan tindakan pengujian pada kolom nim pada pengujian 2 dan pengujian 4 yang membedakan adalah jumlah karakter pengujian. Adapun hasil pengujian 2 dan 3 dapat dilihat pada gambar 16 dan 17 berikut :

```
akademik=# INSERT INTO mahasiswa_nama (nama) VALUES ('Huruf Angka 1234 Spasi');
INSERT 0 1
```

Gambar 16. Pengujian 2 Kolom nama: Bernilai karakter huruf, angka dan spasi

```
akademik=# INSERT INTO mahasiswa_nama (nama) VALUES ('Huruf Angka 1234 Simbol !@#$%');
ERROR:  new row for relation "mahasiswa_nama" violates check constraint "check_nama"
DETAIL:  Failing row contains (Huruf Angka 1234 Simbol !@#$%).
```

Gambar 17. Pengujian 3 kolom nama: Tidak menggunakan karakter simbol

Gambar 16 dan gambar 17 menunjukkan hasil pengujian 2 dan 3 untuk kolom nama, dapat dilihat pada pengujian 2 diinputkan nilai berupa karakter huruf, angka dan spasi pada kolom nama dan hasilnya adalah nilai tersebut diterima oleh sistem basisdata, hal ini sesuai dengan aturan data yang telah ditetapkan. Sedangkan untuk pengujian 3 dapat dilihat bahwa sistem basisdata menolak data yang diinputkan hal ini disebabkan karena nilai yang diinputkan memiliki karakter simbol. Pengujian selanjutnya dilakukan pada kolom tempat_lahir, adapun aturan data pada kolom tersebut adalah 1) tempat lahir maksimal 35 karakter, 2) hanya menggunakan karkater huruf, angka dan spasi, 3) diperbolehkan menggunakan karakter simbol garis mendatar (-), 4) tidak bernilai NULL. Untuk pengujian kolom tempat_lahir yang akan dilakukan uji coba adalah pengujian 3 sedangkan untuk pengujian 1, 2 dan 4 tidak dilakukan uji coba karena pengujian tersebut serupa dengan pengujian sebelumnya. Adapun hasil pengujian 3 untuk kolom tempat_lahir dapat dilihat pada gambar 18 dan gambar 19 berikut:

```
akademik=# INSERT INTO mahasiswa_tempat_lahir (tempat_lahir) VALUES ('Jawa-Timur');
INSERT 0 1
```

Gambar 18. Pengujian 3.1 kolom tempat_lahir: Penggunaan simbol garis mendatar (-)

```
akademik=# INSERT INTO mahasiswa_tempat_lahir (tempat_lahir) VALUES ('Jawa_Timur');
ERROR:  new row for relation "mahasiswa_tempat_lahir" violates check constraint "check_tempat_lahir"
DETAIL:  Failing row contains (Jawa_Timur).
```

Gambar19. Pengujian 3.2. kolom tempat_lahir: Penggunaan simbol selain garis mendatar (-)

Untuk pengujian kolom tempat_lahir dilakukan dua tahap pengujian yang ditunjukkan pada gambar 18 dan gambar 19. Gambar 18 menunjukkan hasil pengujian penginputan data dengan nilai yang memiliki karakter garis mendatar (-), hasil pengujian menunjukkan data yang diinputkan diterima oleh sistem basis data. Sedangkan gambar 19 menunjukkan hasil pengujian penginputan data dengan nilai yang memiliki karakter selain garis mendatar (-), hasil pengujian menunjukkan data yang diinputkan tidak diterima oleh sistem basisdata, hal ini sesuai dengan aturan data untuk kolom tempat_lahir yang menyatakan bahwa diperbolehkan menggunakan karakter garis mendatar (-).

Pengujian berikutnya dilakukan pada kolom tanggal_lahir, adapun aturan data pada kolom tanggal_lahir adalah: 1) tidak bernilai NULL, 2) umur harus diatas 18 tahun. Untuk pengujian 1 tidak akan dilakukan uji coba karena pengujian dengan skenario serupa telah dilakukan pada tahap sebelumnya. Untuk pengujian 2 akan dilkakukan dua kali uji coba. Adapun hasil pengujian ditunjukkan pada gambar 20 dan gambar 21 berikut:

```
akademik=# INSERT INTO mahasiswa_tanggal_lahir (tanggal_lahir) VALUES ('2001-08-22');
INSERT 0 1
```

Gambar 20. Pengujian 2.1 kolom tanggal_lahir: Penginputan umur diatas 18 tahun

```
akademik=# INSERT INTO mahasiswa_tanggal_lahir (tanggal_lahir) VALUES ('2005-08-22');
ERROR:  new row for relation "mahasiswa_tanggal_lahir" violates check constraint "check_tanggal_lahir"
DETAIL:  Failing row contains (2005-08-22).
```

Gambar 21. Pengujian 2.2 kolom tanggal_lahir: Penginputan umur dibawah 18 tahun

Gambar 20 dan gambar 21 menunjukkan hasil pengujian kolom tanggal_lahir. Pada gambar 20 data yang diinputkan adalah data tanggal dengan tahun 2001 yang berarti berumur 19 tahun dan ditunjukan bahwa sistem basisdata menerima data tersebut. Berbeda dengan gambar 21 yang menunjukkan sistem basisdata menolak data yang diinputkan, hal ini

dikarenakan tahun tanggal yang diinputkan adalah 2005 yang berarti berumur 15 tahun.

Pengujian selanjutnya dilakukan pada kolom gender, kolom gender merupakan kolom yang akan menampung data gender (laki laki / perempuan). Aturan data pada kolom gender adalah: 1) kolom gender hanya bernilai 1 karakter huruf, L yang berarti laki laki dan P yang berarti perempuan, 2) tidak bernilai NULL. Untuk pengujian 2 tidak akan dilakukan uji coba karena telah dilakukan pengujian serupa telah dilakukan pada tahap sebelumnya. Adapun pengujian untuk kolom gender ditunjukkan pada gambar 22 berikut:

```
akademik=# INSERT INTO mahasiswa_gender (gender) VALUES ('S');
ERROR:  new row for relation "mahasiswa_gender" violates check constraint "check_gender"
DETAIL:  Failing row contains (S).
```

Gambar 22. Pengujian 1 kolom gender: Hanya bernilai L dan P

Gambar 22 merupakan pengujian kolom gender, dapat dilihat pada kolom gender diinputkan data dengan nilai yang tidak sesuai dengan aturan data sehingga sistem basisdata tidak menerima inputan data tersebut.

Kolom selanjutnya yang akan diuji coba adalah kolom kode_mk. Kolom kode_mk hanya memiliki satu aturan data yaitu data pada kode_mk sama dengan data pada kolom kode_mk pada tabel mahasiswa, yang berarti kolom tersebut saling berelasi. Adapun hasil pengujian untuk kolom kode_mk dapat dilihat pada gambar 23 berikut:

```
akademik=# INSERT INTO mahasiswa_kode_mk (kode_mk)
akademik-# VALUES ('SBD02');
ERROR:  insert or update on table "mahasiswa_kode_mk" violates foreign key constraint
"mahasiswa_kode_mk_kode_mk_fkey"
DETAIL:  Key (kode_mk)=(SBD02) is not present in table "matakuliah_kode_mk".
```

Gambar 23. Pengujian 1 kolom kode_mk: Penambahan data yang tidak ada pada kolom relasi

Gambar 23 merupakan hasil penginputan data pada kolom kode_mk, dapat dilihat bahwa data yang diinputkan tidak diterima oleh sistem basisdata, hal disebabkan karena data yang diinputkan tidak terdapat pada kolom yang berelasi dalam kasus ini adalah kolom kode_mk pada tabel matakuliah_kode_mk. Gambar 24 berikut menunjukkan data pada kolom kode_mk pada tabel matakuliah_kode_mk:

```
akademik=# SELECT * FROM matakuliah_kode_mk ;
 kode_mk
---------
 SBD01
 JKOM2
 ALPRO
 ANSI1
(4 rows)
```

Gambar 24. Data pada tabel matakuliah_kode_mk

Pada gambar 24 dapat dilihat pada kolom kode_mk tabel matakuliah_kode_mk terdapat empat baris data kode matakuliah yaitu SBD01, JKOM2, ALPRO dan ANSI1 yang berarti hanya data tersebut yang dapat diinputkan kedalam tabel yang berelasi. Pada gambar 23 ditunjukkan bahwa data yang diinputkan pada kolom kode_mk adalah SBD02, data tersebut tidak terdapat pada kolom kode_mk tabel matakuliah_kode_mk yang menjadi relasi dari kolom tersebut sehingga data tersebut tidak diterima oleh sistem basisdata. Penggunaan *constraint* ini menjamin adanya keabsahan data, dimana dalam kasus ini matakuliah yang akan dipilih oleh mahasiswa adalah matakuliah yang terdaftar pada tabel matakuliah.

Pembahasan sebelumnya telah dilakukan adalah penamabahan data dengan memasukkan data yang tidak sesuai dengan aturan data. Selanjutnya adalah melakuakan pengujian penginputan data yang sesuai dengan aturan data. Pada tahap ini akan dilakukan penambahan data pada tabel matakuliah dan tabel mahasiswa seperti yang ditunjukkan pada gambar 1 dan gambar 2. Adapun hasil pengujian tersebut dapat dilihat pada gambar 25 dan gambar 26 berikut:

```
akademik=# INSERT INTO matakuliah
akademik-# (kode_mk,nama_mk,sks) VALUES
akademik-# ('SBD01','Sistem Basis Data',3),
akademik-# ('JKOM2','Jaringan Komputer 2',3),
akademik-# ('ALPRO','Algoritma Pemrograman',4),
akademik-# ('ANSI1','Analisis Perancangan Sistem',4);
INSERT 0 4
```

Gambar 25. Penambahan data pada tabel matakuliah

```
akademik=# INSERT INTO mahasiswa
akademik-# (nim,nama,tempat_lahir,tanggal_lahir,gender,kode_mk) VALUES
akademik-# ('20202208103','Reza Maulana','Makassar','2001-08-22','L','SBD01'),
akademik-# ('20202308104','Namashita Haida','Suli-Luwu','2000-08-23','P','ALPRO'),
akademik-# ('20202408105','Nizhamul Haq','Yogyakarta','1999-08-24','L','JKOM2');
INSERT 0 3
```

Gambar 26. Penambahan data pada tabel mahasiswa

Gambar 25 dan gambar 26 menunjukkan hasil pengujian penambahan data pada tabel matakuliah dan tabel mahasiswa, dapat dilihat bahwa data yang diinputkan berhasil diterima oleh sistem basisdata karena data yang diinputkan sesuai dengan aturan data

KESIMPULAN

Penggunaan *constraint* pada *basisdata* memberikan banyak keuntungkan, salah satunya adalah pembatasan pengiputan. Dengan menggunakan *constraint*, *end-user basisdata* akan dipaksa memasukkan data yang sesuai dengan aturan yang telah ditetapkan. Pada penelitian ini dari semua skenario yang ditentukan pada tabel 1 dapat dilihat bahwa secara keseluruhan skenario tersebut dapat diterapkan dengan menggunakan *constraint*.

Penulis sadar masih sangat banyak kekurangan pada penelitian ini, hal ini merupakan keterbatasan penulis baik itu keterbatasan pengetahuan dan keterbatasan waktu. Pada penelitian ini yang dilakukan adalah pembatasan penginputan data pada level basisdata, namun pembatasan juga dapat dilakukan pada level bahasa pemrograman ini adalah peluang penelitian berikutnya untuk menguji efesiensi dan efektifitas pembatasan penginputan pada level basisdata dengan level bahasa pemrograman.

DAFTAR PUSTAKA

Darmanto, E. (2015). Analisa Optimalisasi Bahasa SQL Berdasarkan Relational Algebra pada Kasus Rekapitulasi Mahasiswa Layak Wisuda. *Jurnal SIMETRIS*, *6*(2). https://jurnal.umk.ac.id/index.php/simet/article/view/479/514

Hartono, N., Utami, E., & Amborowati, A. (2016). Migrasi dan Optimalisasi Database Sistem Informasi Manajemen Universitas Cokroaminoto Palopo. *Jurnal Buana Informatika*, *7*(4), Article 4. https://doi.org/10.24002/jbi.v7i4.766

Raharjo, S. (2012). *Constraint Basisdata Sebagaia Fondasi yang Kuat Dalam Pengembangan Sistem Informasi*. 6.

Raharjo, S. (2013). *Integrity Constraint Basisdata Relasional dengan Menggunakan PL/PGSQL dan Check Constraint*. 6.

Raharjo, S., & Istiyanto, J. (2003). *Keamanan Akses ke PostgreSQL Melalui PHP (Menggunakan Apache Web Server pada GNU/Linux)*. Penerbit Andi.

Rob, P., & Coronel, C. (2009). *Database Systems: Design, Implementation, And Management.* Course Technology.

Setyanto, A. E., & Setyanto, A. E. (2013). Memperkenalkan Kembali Metode Eksperimen dalam Kajian Komunikasi. *Jurnal ILMU KOMUNIKASI*, *3*(1), Article 1. https://doi.org/10.24002/jik.v3i1.239

Utami, E., & Raharjo, S. (2006). RDBMS dengan PostgreSQL di GNU. *Linux, Andi Offset, Yogyakarta*.