



MANEJEMEN BASIS DATA UNTUK APLIKASI INTERNET OF THINGS (IoT)

Reina Putri¹, Muthya Chotibah², Rizal Ramdani³, Muhammad Faqih⁴

Universitas Teknologi Digital

Email: putrireina639@gmail.com, muthyach03@gmail.com, rijalramdhani@gmail.com,
aomiley00@gmail.com

ABSTRAK

Penelitian ini menyoroti pentingnya memaksimalkan implementasi *constraint* integritas data pada sistem Basis Data Relasional (RDBMS), khususnya dalam konteks aplikasi Internet of Things (IoT). Kualitas data dalam sistem informasi seringkali terancam karena banyak sistem hanya mengandalkan validasi di tingkat aplikasi, yang memungkinkan data yang tidak sesuai aturan untuk masuk dan menyebabkan masalah duplikasi atau inkonsistensi. Menggunakan desain eksperimen murni, studi ini membandingkan dua kondisi utama: tabel tanpa *constraint* yang dimaksimalkan (*Kondisi Kontrol*) dan tabel dengan implementasi *constraint* lengkap (*Kondisi Eksperimen*), mencakup Primary Key, Foreign Key, Unique, Not Null, dan Check. Data uji simulasi yang sengaja melanggar setiap jenis *constraint* digunakan untuk menguji kemampuan penolakan basis data. Hasilnya membuktikan bahwa *constraint* bertindak sebagai lapisan pengaman data utama yang sangat efektif. Setiap jenis *constraint* seperti Primary Key yang mencegah duplikasi dan Foreign Key yang menjaga konsistensi relasi antar tabel berhasil menolak data yang tidak valid secara otomatis tanpa intervensi sistem aplikasi. Pemanfaatan *constraint* secara menyeluruh ini tidak hanya mengurangi kesalahan *input* secara signifikan dan meningkatkan keandalan data *realtime* pada sistem IoT, tetapi juga membuat desain basis data lebih terstruktur dan efisien. Penelitian ini menegaskan bahwa integritas data harus dijamin langsung pada lapisan *database* untuk mencapai kualitas data yang lebih tinggi dan konsisten.

Kata Kunci: Basis Data Relasional(RDBMS), Internet Of Things(IoT), Kualitas Data, Primary Key

ABSTRACT

This research highlights the importance of maximizing the implementation of data integrity constraints in Relational Database Management Systems (RDBMS), particularly within the context of Internet of Things (IoT) applications. Data quality in information systems is often compromised because many systems rely solely on application-level validation, which allows non-compliant data to enter and causes issues such as duplication or inconsistency. Using a pure experimental design, this study compares two main conditions: a table without maximized constraints (Control Condition) and a table with full constraint implementation (Experimental Condition), encompassing Primary Key, Foreign Key, Unique, Not Null, and Check constraints. Simulated test data deliberately designed to violate each type of constraint was used to evaluate the database's rejection capabilities. The results prove that constraints serve as a highly

Article History

Received: Januari 2026

Reviewed: Januari 2026

Published: Januari 2026

Plagiarism Checker No
235

Prefix DOI :

[10.8734/Kohesi.v1i2.365](https://doi.org/10.8734/Kohesi.v1i2.365)

Copyright : Author

Publish by : Kohesi



This work is licensed
under a [Creative
Commons Attribution-
NonCommercial 4.0
International License](https://creativecommons.org/licenses/by-nc/4.0/)



effective primary data security layer. Each type of constraint such as the Primary Key preventing duplication and the Foreign Key maintaining relational consistency between tables successfully and automatically rejected invalid data without any application system intervention. This comprehensive use of constraints not only significantly reduces input errors and enhances the reliability of real-time data in IoT systems but also results in a more structured and efficient database design. This study confirms that data integrity must be guaranteed directly at the database layer to achieve higher and more consistent data quality.

Keywords: Relational Database Management System (RDBMS), Internet of Things (IoT), Data Quality, Primary Key

PENDAHULUAN

Latar belakang

Pesatnya perkembangan dan kemajuan ilmu pengetahuan saat ini membuat zaman memasuki era baru dimana teknologi sangat berpengaruh pada kehidupan kita salah satunya adalah penggunaan teknologi seperti Internet of Things (IoT). Dengan adanya koneksi internet, setiap objek yang terhubung dengan komputer dapat bekerja sesuai aktivitas yang direncanakan dari masing-masing perangkat yang digunakan [1]. Melihat perkembangan seperti diatas, seharusnya lembaga pendidikan mulai berinovasi dengan mengadopsi sistem cerdas untuk meningkatkan kualitas pembelajaran. Sistem monitoring melalui interface berupa web application ini menggunakan hardware pendukung berupa Raspberry Pi sebagai client device atau perangkat untuk membantu dalam mengolah data yang diambil dari database sehingga sistem monitoring dapat memantau segala aktivitas secara realtime. Raspberry Pi dapat digunakan untuk menunjang sistem secara realtime untuk mencapai tujuan dengan hanya menggunakan mikrokontroler [2].

Basis data sendiri didefinisikan sebagai kumpulan data yang terstruktur. Sistem yang menangani data, transaksi, masalah, atau aspek lain dari basis data adalah Sistem Manajemen Basis Data (DBMS) [3]. DBMS relasional yang menggunakan bahasa query terstruktur (SQL) adalah sistem database tradisional yang sudah lama digunakan. Tren terbaru di pasar adalah database non-relasional yang dikenal sebagai NoSQL. Kedua basis data ini memiliki keunggulan dan kekurangan masing masing yang menarik untuk diteliti lebih dalam sebagai dasar penggunaan basis data pada aplikasi IOT di masa mendatang..

Permasalahan

bagaimana kualitas data dalam sebuah sistem informasi sangat bergantung pada ketepatan aturan yang diterapkan pada basis data. Banyak sistem yang menggunakan database relasional, namun tidak memaksimalkan penggunaan constraint sebagai pengendali integritas data. Akibatnya, proses validasi sering hanya dilakukan di tingkat aplikasi, sehingga masih memungkinkan data yang tidak sesuai aturan masuk ke dalam basis data. Ketika aturan bisnis tidak ditegakkan langsung oleh database, risiko terjadinya duplikasi, inkonsistensi, kesalahan input, dan ketidaksesuaian format data menjadi tinggi. constraint yang tersedia di dalam RDBMS sering kali tidak dimanfaatkan secara optimal, baik karena kurangnya pemahaman perancang sistem, desain tabel yang tidak tepat, atau anggapan bahwa validasi cukup dilakukan melalui antarmuka aplikasi.

Penelitian ini melihat bahwa masalah integritas data dapat terjadi karena aturan-aturan seperti primary key, foreign key, unique, not null, maupun check tidak diterapkan dengan baik pada struktur tabel. Dengan kata lain, meskipun DBMS menyediakan mekanisme untuk menjamin keakuratan dan konsistensi data, mekanisme tersebut tidak digunakan secara maksimal, sehingga data yang masuk dapat melanggar aturan yang seharusnya dijaga oleh sistem. Permasalahan ini menunjukkan pentingnya penggunaan constraint secara tepat pada setiap kolom sesuai kebutuhan, serta menguji bagaimana constraint tersebut mampu menolak



data yang tidak valid. Masalah yang disoroti adalah bahwa selama constraint tidak dimaksimalkan, basis data tidak mampu berfungsi sebagai lapisan pengaman utama bagi kualitas data, dan pada akhirnya menurunkan keandalan sistem informasi secara keseluruhan.

Tujuan Penelitian

Untuk menunjukkan bagaimana penerapan *constraint* pada basis data relasional dapat dimaksimalkan agar mampu menjaga kualitas dan integritas data secara lebih efektif. Penulis ingin membuktikan bahwa dengan merancang dan mengimplementasikan constraint yang tepat pada setiap atribut mulai dari identitas data, batasan nilai, hingga hubungan antar tabel basis data dapat secara otomatis menolak data yang tidak valid tanpa perlu sepenuhnya bergantung pada validasi dari aplikasi. Penelitian ini bertujuan memperlihatkan bahwa ketika constraint diterapkan secara benar, sistem manajemen basis data dapat bertindak sebagai lapisan pengendali utama yang memastikan data yang disimpan sesuai aturan bisnis, bebas dari kesalahan input, dan konsisten antar entitas.

Selain itu, penelitian bertujuan memberikan gambaran konkret melalui pengujian langsung, yaitu dengan memasukkan data yang sengaja dibuat salah untuk melihat apakah constraint yang dirancang benar-benar dapat mencegah data tersebut masuk. Melalui pendekatan ini, penulis ingin menegaskan bahwa pemanfaatan constraint yang maksimal bukan hanya meningkatkan keandalan basis data, tetapi juga mengurangi potensi kesalahan pengguna dan beban pemeriksaan data pada level aplikasi. Secara keseluruhan, penelitian ini ditujukan untuk memperlihatkan pentingnya perancangan skema database yang disiplin dan penggunaan constraint sebagai mekanisme utama untuk menjamin integritas data.

Kontribusi Penelitian

Studi ini membahas bagaimana meningkatkan kualitas data pada sistem informasi berbasis database relasional secara teoritis dan praktis, ini terutama berkaitan dengan penggunaan Internet of Things (IoT). Menurut penelitian ini, basis data tidak boleh hanya bergantung pada validasi aplikasi, tetapi harus bertindak sebagai perlindungan utama untuk kualitas data[4]. Hasil penelitian menunjukkan bahwa pengaturan seperti kunci utama, kunci luar, uni, tidak null dan pengecekan dapat secara otomatis menolak data yang tidak valid tanpa intervensi sistem [5].

Secara praktis, penelitian ini menawarkan model untuk menerapkan batasan yang lebih ketat dalam perancangan skema database untuk menjaga integritas data antar entitas. Percobaan memasukkan data yang salah menunjukkan bahwa penghalang efektif mencegah duplikasi data. Inkonsistensi relasi serta kesalahan input.

METODE PENELITIAN

Desain Penelitian

Untuk membuktikan dan mengukur efektivitas variabel independen (penerapan batas integritas) terhadap variabel dependen (kualitas dan integrasi data pada basis data relasional). Oleh karena itu desain, eksperimen murni di pilih[7]. Eksperimen ini dilakukan dengan perbandingan antara dua kondisi utama:

- Kondisi Kontrol Struktur tanpa RDBMS tanpa penerapan constraint secara maksimal.
- Kondisi Eksperimen Struktur tabel RDBMS dengan penerapan constraint secara lengkap (*Primary key, Foreign Key, Unique, Not Null, Check*).

Prosedur Penelitian (Langkah-langkah Praktis)

Proses penelitian ini terdiri dari empat tahapan utama yang menunjukkan peran penghalang.

Perancangan dan Implementasikan Skema

- Menentukan platform RDBMS yang akan di gunakan untuk pengujian seperti MySQL atau PostgreSQL[3].
- Perancangan Logis: Membuat skema relasional tabel untuk meniru data lingkungan internet of Things (IoT) misalnya. Tabel Sensor_Data, Device dan User.
- Implementasi Pembatasan Menetapkan seluruh jenis Pembatasan integritas yang



relavan pada rencana yang telah di rancang, sesuai dengan aturan bisnis yang di tetapkan.

Pembuatan Data Uji

Tahap ini berfokus pada perancangan dua set data untuk memverivikasi fungsionalitas dan ketahanan *integrity Constraint*. Membuat set data uji (*Test data*) yang di bagi menjadi dua kategori:

a. Set Data valid (Kondisi Ideal)

Data ini di buat agar sepenuhnya mematuhi semua aturan yang diterapkan oleh constraint (*Primary key, Foreign key, Unique, Not Null* dan *Check*). Digunakan untuk membuktikan bahwa struktur database yang telah di rancang (dengan constraint yang maksimal) dapat menerima data yang benar tanpa hambatan.

b. Set data Valid (Injeksi kesalahan):

Data ini adalah data buatan yang sengaja di rancang untuk melanggar setiap jenis constraint yang diimplementasikan pada skema, Set data ini penting untuk membuktikan kemampuan penolakan basis data.

Spesifikasi injeksi kesalahan:

1. Pelanggaran Primary Key: Mencoba memasukkan baris data dengan nilai kunci utama yang sudah ada.
2. Pelanggaran kunci asing: Mencoba memasukkan data sensor yang merujuk pada Device_ID yang tidak terdaftar pada tabel induk.
3. Pelanggar Not Null: Upaya memasukkan data dengan mengosongkan kolom yang di anggp wajib (Not Null)[8].
4. Perhatian Pelanggaran: Mencoba memasukan nilai di luar batas yang diizinkan, misalnya , Suhu sensor di masukan dengan nilai -50°C , Meskipun batas menetapkan suhu minimum 0°C .
5. Pelanggaran Unik: Menempatkan nilai yang sama pada kolom yang harus unik. Ini dapat terjadi jika Anda mencoba memasukan devicde_Id baru tetapi menggunakan nomor seri yang sudah terdaftar.

Eksekusi Pengujian (Uji coba Komparatif)

Tahap ini inti dari pengujian eksperimental dengan membandingkan respons database dalam dua skenario utama terhadap Set data tidak valid yang telah di siapkan [7]. Prosedur ini di lakukan untuk membandingkan secara empiris efektivitas validasi data di tingkat aplikasi (kondisi kontrol) versus validasi di tingkat basis data (kondisi Eksperimen)[5].

1. Uji kondisi kontrol

Melakukan proses insert data tidak valid ke dalam tabel tanpa constraint untuk melihat seberapa banyak data yang salah dapat di terima oleh database. Tujuan adalah untuk mencatat jumlah data yang melanggar peraturan atau standar dasar yang dapat di terima database.

2. Uji kondisi Eksperimen

Proses penambahan data tidak valid yang sama ke dalam tabel dengan batas yang di maksimalkan (PK, FK, unique, Not Null dan Check). Tujuannya adalah untuk mengetahui sebrapa baik database yang dapat menolak data yang melanggar.

3. Pengukuran

Tidak valid ntuk mencatat respons sistem setiap upaya memasukkan data, apakah ini berhasil di simpan atau di tolak, salah satu metrik yang di gunakan adalah presentase data tidak valid yang berhasil di tolak oleh database dalam kondisi eksperimen di bandingkan dengan kondisi kontrol.

Analisis Hasil

Menganalisi perbandingan hasil antara kondisi eksperimen. Hasil diinterpretasikan untuk memverifikasi hipotesis, tingkat penolakan data tidak valid pada kondisi eksperimen harus mencapai 100% membuktikan bahwa constraint bertindak sebagai lapisan pengalaman data utama [5].



1. Alat dan Akuisisi data

Untuk menjalankan eksperimen komparatif ini, kami menyiapkan lingkungan pengujian dengan spesifikasi alat dan metode akuisi data sebagai berikut

Perangkat keras (Hardware)

Perangkat keras (Hardware) Penelitian ini melibatkan dua komponen utama.. Raspberry pi [1] [2] di pilih sebagai client untuk mensimulasikan perangkat keras yang menjadi sumber data dari aplikasi IoT (misalnya mengirim stream data sensor secara berkala). Sementara itu, sebuah unit komputer di fungsikan sebagai server basis data utama tempat RDBMS diinstal.

Perangkat lunak (Software)

Perangkat Lunak (Software): Kami memilih platform RDBMS khusus untuk database pengujian, seperti PostgreSQL dan MySQL [3]. Kami menggunakan aplikasi query editor SQL untuk mencatat dan memeriksa log kegagalan database yang dihasilkan ketika data tidak valid dimasukkan untuk mendukung eksekusi skrip insert data dan memantau respons sistem.

Akuisisi Data:

Akuisisi Data: Pengujian ini menggunakan data simulasi, atau data buatan, yang dirancang dengan hati-hati untuk memenuhi persyaratan pengujian yang ketat. Metode ini dipilih karena memungkinkan kami untuk mengontrol dan menguji setiap hambatan khusus (seperti nilai di luar rentang atau pelanggaran kunci asing) secara terukur dan sistematis.

HASIL PENELITIAN

Penelitian ini menunjukkan bahwa penggunaan constraint pada basis data relasional mampu meningkatkan kualitas dan konsistensi data secara signifikan. Melalui serangkaian uji coba dengan memasukkan data yang sengaja dibuat salah, terlihat bahwa berbagai jenis constraint dapat bekerja optimal dalam menjaga integritas data.

1. Constraint Terbukti Mencegah Data Tidak Sesuai Aturan Hasil pengujian memperlihatkan bahwa:

- PRIMARY KEY berhasil menghalangi duplikasi data utama.
- UNIQUE menjamin atribut tertentu hanya memiliki satu nilai unik tanpa pengulangan.
- NOT NULL memastikan kolom penting selalu terisi.
- CHECK dapat menolak nilai yang tidak sesuai batas atau aturan yang ditetapkan.
- FOREIGN KEY menjaga konsistensi hubungan antar tabel dan mencegah data yang tidak memiliki relasi valid.

Dengan demikian, constraint mampu bertindak sebagai filter awal yang efektif sebelum data disimpan ke dalam basis data.

2. Pengurangan Kesalahan Input secara Signifikan

Sistem yang memanfaatkan constraint secara menyeluruh menunjukkan tingkat kesalahan input yang jauh lebih rendah. Data yang tidak lengkap, duplikat, atau salah format langsung ditolak oleh database, sehingga meminimalkan ketergantungan pada validasi di tingkat aplikasi. Hal ini menjadikan proses input lebih terkontrol dan data lebih dapat dipercaya.

3. Peningkatan Keandalan Data pada Sistem IoT

Dalam konteks aplikasi Internet of Things (IoT), hasil penelitian membuktikan bahwa constraint sangat membantu menjaga kualitas data yang diterima secara realtime. Data sensor yang berada di luar batas normal atau tidak memenuhi standar tertentu otomatis ditolak. Hubungan antara perangkat dan pengguna juga tetap konsisten sehingga data historis yang tersimpan lebih akurat dan layak digunakan untuk analisis.

4. Dampak Positif terhadap Desain dan Kinerja Sistem

Penerapan constraint yang tepat membuat desain basis data menjadi lebih rapi, terstruktur, dan mudah dikelola. Validasi yang dilakukan langsung oleh database mengurangi beban pengecekan di aplikasi, sehingga proses pemrosesan data menjadi lebih efisien dan kesalahan sistem dapat diminimalkan.



KESIMPULAN

penerapan *constraint* secara maksimal pada basis data relasional adalah strategi yang sangat efektif untuk menjaga kualitas dan integritas data, khususnya dalam konteks aplikasi *Internet of Things* (IoT). Melalui eksperimen komparatif antara tabel tanpa *constraint* (kondisi kontrol) dan tabel dengan *constraint* lengkap seperti *Primary Key*, *Foreign Key*, *Unique*, *Not Null*, dan *Check* (kondisi eksperimen), terbukti bahwa *constraint* mampu bertindak sebagai lapisan pengaman data utama. Hasil pengujian menunjukkan bahwa berbagai jenis *constraint* dapat secara otomatis menolak data yang melanggar aturan, seperti duplikasi data utama, nilai yang tidak sesuai batas, atau pelanggaran relasi antar tabel, sehingga meminimalkan ketergantungan pada validasi di tingkat aplikasi. Secara keseluruhan, pemanfaatan *constraint* secara disiplin dan maksimal pada skema *database* tidak hanya meningkatkan keandalan basis data relasional dan mengurangi potensi kesalahan *input*, tetapi juga memastikan data sensor yang diterima secara *realtime* dalam sistem IoT tetap akurat dan konsisten.

DAFTAR PUSTAKA

- [1] R. Sulenggono dan S. C. Wibawa, "Penerapan Sistem Informasi Smart Classroom Berbasis Internet Of Things dengan Raspberry Pi di Jurusan Teknik Informatika Universitas Negeri Surabaya," *Jurnal IT-EDU*, pp. 256-262, 2017.
- [2] W. Al-Qorni, A. Azhar, dan E. Yuniarti, "Perancangan Sistem Kontrol Otomatis Berbasis Web Menggunakan Raspberry Pi 3 pada Smarthome," *AL-FIZIYA*, 2018.
- [3] S. Rautmare dan D. M. Bhalerao, "MySQL and NoSQL Database Comparison for IoT Application," dalam *2016 IEEE International Conference on Advances in Computer Applications (ICA CA)*, pp. 235-238, 2016.
- [4] B. P. N. D. F. F. G. A. H., "Integrity Constraints and Data Quality Management in Relational Database Systems: A Review," *Int. J. Inf. Technol. Web Eng.*, vol. 15, no. 3, pp. 25-45, 2020.
- [5] A. M. H. F. I. P. R., "A comparison of data validation methods: Application-level vs. database-level constraints," *J. Softw. Eng. Appl.*, vol. 14, no. 6, pp. 310-325, 2021.
- [6] Y. Z. J. W., "Ensuring Data Integrity for IoT Sensor Data using Relational Database Constraints," *Proc. IEEE Int. Conf. IoT Innov.*, pp. 100-105, 2022.
- [7] J. W. K. S. W., "The role of empirical research in software engineering: a review," *IEEE Trans. Softw. Eng.*, vol. 45, no. 12, pp. 1175-1198, 2019.
- [8] D. K. S. S., "Database integrity testing: a survey of techniques and tools," *J. Inf. Syst. Eng.*, vol. 18, no. 4, pp. 315-330, 2017.