

Tugas Besar Basis Data
Perancangan dan Pemodelan Basis Data Produk
di Toko BlissBlocks

Dosen Pengampu : Rima Dias Ramadhani S.Kom.,M.Kom.



Disusun oleh:

Muhammad Naufal Farabbi	(2211110013)
Muhammad Fadhlalhifan Siregar	(2211110016)
Andi Hisyam F. F	(2211110017)
Muhammad Ihsan Prawira Hutomo	(2211110022)
Hanif Fakhri Pratama	(2211110037)

PROGRAM STUDI S1 SAINS DATA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2023

Daftar Isi

Bab I Pendahuluan	4
1.1. Latar Belakang	4
1.2. Tujuan.....	5
1.3. Manfaat.....	5
Bab II Dasar Teori.....	6
2.1 Basis Data.....	6
2.1.1 Pengertian Basis Data	6
2.1.2 Komponen Basis Data.....	6
2.1.3 Arsitektur Basis Data	7
2.2 Entity Relationship Diagram (ERD)	8
2.2.1 Pengertian ERD	8
2.2.2 Kardinalitas Relasi pada ERD	8
2.3 Normalisasi Tabel.....	9
2.3.1 Bentuk Normal Tahap 1 (1NF)	9
2.3.2 Bentuk Normal Tahap 2 (2NF)	9
2.3.3 Bentuk Normal Tahap 3 (3NF)	10
2.4 Basis Data MySQL.....	10
Bab III Perancangan Basis Data	12
3.1 Tinjauan Instansi	12
3.1.1 Proses Bisnis / Proses Kerja Perusahaan	12
3.2 Perancangan ERD	12
3.2.1 Entitiy Relationship Diagram (ERD)	14
3.2.2 Implementasi ERD ke Basis Data Fisik	15
3.3 Normalisasi Tabel.....	20
3.3.1Bentuk Normal Tahap 1 (1NF)	20
3.3.2 Bentuk Normal Tahap 2 (2NF).....	22
3.3.3 Bentuk Normal Tahap 3 (3NF)	25
Bab IV Implementasi	26
4.1 Implementasi ke Database MySQL.....	26
4.1.1 Struktur Tabel.....	26
4.1.1 Relasi Tabel.....	29
Bab V Penutup	32

5.1	Kesimpulan.....	32
5.2	Saran.....	32
	Daftar Pustaka.....	33

Bab I

Pendahuluan

1.1. Latar Belakang

Dalam era modern, teknologi informasi telah menjadi pilar utama bagi perkembangan berbagai industri, memunculkan kebutuhan mendesak akan pengelolaan data yang efisien dan inovatif. Dalam konteks BlissBlocks, perusahaan mainan konstruksi yang menekankan kegembiraan dan kreativitas, pemanfaatan teknologi canggih seperti desain 3D, analisis tren konsumen, dan optimalisasi rantai pasokan menjadi sangat relevan. Permintaan konsumen untuk pengalaman bermain yang lebih interaktif dan personalisasi menetapkan data sebagai komponen integral untuk memahami preferensi dan perilaku konsumen.

BlissBlocks, sebagai pemimpin di industri mainan konstruksi, berkomitmen untuk memberikan pengalaman bermain yang membangun kreativitas dan kegembiraan. Dengan koleksi produk inovatif dan pendekatan desain yang unik, perusahaan ini telah memperoleh reputasi yang kuat. Didukung oleh tim kreatif dan tekun, BlissBlocks terus berinovasi untuk memenuhi harapan konsumen yang terus berkembang.

Meskipun masih dalam tahap pertumbuhan awal, BlissBlocks menghadapi tantangan dalam mengelola volume data yang masih terbatas namun semakin berkembang. Sebagai perusahaan yang baru tumbuh, sumber data seperti feedback konsumen, desain produk, dan informasi produksi masih terbatas, namun memiliki potensi besar untuk memberikan wawasan yang berharga. Integrasi data yang hati-hati menjadi langkah krusial untuk memastikan pemahaman menyeluruh terhadap tren awal dan preferensi konsumen yang mungkin terus berubah seiring waktu.

Keamanan data, keakuratan inventaris, dan pelacakan tren konsumen tetap menjadi aspek-aspek penting, bahkan pada tahap awal pertumbuhan BlissBlocks. Sebagai respons terhadap skala bisnis yang relatif kecil, solusi pengelolaan data yang fleksibel dan dapat disesuaikan menjadi fokus utama. BlissBlocks perlu memastikan bahwa perancangan basis data tidak hanya memenuhi kebutuhan saat ini tetapi juga dapat berkembang seiring dengan ekspansi perusahaan.

Oleh karena itu, perancangan database yang kokoh dan efisien menjadi kunci keberlanjutan dan pertumbuhan BlissBlocks. Melalui perancangan database yang terstruktur,

BlissBlocks dapat memastikan akses yang cepat dan akurat terhadap data, memperkuat keamanan informasi, dan mendukung pengambilan keputusan yang lebih baik. Dengan merancang basis data yang adaptif dan terintegrasi, BlissBlocks dapat memaksimalkan potensi kreatifnya, meningkatkan efisiensi operasional, dan memberikan pengalaman bermain yang lebih memuaskan bagi pelanggan.

1.2. Tujuan

Tujuan utama BlissBlocks adalah memberikan pengalaman bermain yang kreatif dan memuaskan bagi anak-anak di seluruh dunia. Dengan koleksi produk inovatif dan pendekatan desain yang unik, BlissBlocks telah memperoleh reputasi sebagai pemimpin di industri mainan konstruksi. Perusahaan ini berkomitmen untuk terus berinovasi guna memenuhi harapan konsumen yang berkembang. Selain itu, BlissBlocks memiliki tujuan untuk menjadi pemimpin di industri ini, dengan fokus pada pengembangan produk-produk berkualitas tinggi yang mengakomodasi perkembangan teknologi terbaru. BlissBlocks juga menetapkan tujuan untuk meningkatkan kepuasan pelanggan melalui layanan pelanggan yang baik dan mendengarkan dengan seksama umpan balik pelanggan.

1.3. Manfaat

BlissBlocks, dengan fokusnya pada mainan konstruksi, memberikan manfaat nyata dalam perkembangan anak-anak. Produk-produk inovatif dan desain yang unik tidak hanya merangsang kreativitas dan imajinasi, tetapi juga mendukung pengembangan kognitif. Saat anak-anak bermain dengan balok-blokannya, mereka terlibat dalam aktivitas yang memerlukan perencanaan, pemikiran spasial, dan pemecahan masalah, memberikan dampak positif pada perkembangan otak dan keterampilan motorik halus. Selain itu, BlissBlocks juga berperan dalam membentuk keterampilan sosial, karena anak-anak belajar berkolaborasi dan berkomunikasi saat bermain bersama. Dengan cara ini, BlissBlocks bukan hanya menyediakan hiburan, tetapi juga memberikan kontribusi positif pada pembelajaran dan pertumbuhan anak-anak.

Bab II

Dasar Teori

2.1 Basis Data

2.1.1 Pengertian Basis Data

Basis data merupakan kumpulan data yang tersusun secara sistematis dan terstruktur dalam suatu perangkat penyimpanan elektronik. Jenis data yang dapat disimpan dalam basis data melibatkan berbagai format, seperti teks, angka, gambar, video, dan file. Untuk mengelola data ini, organisasi umumnya menggunakan Sistem Manajemen Basis Data (DBMS) yang bertugas menyimpan, mengambil, dan mengedit informasi dalam basis data tersebut. Pentingnya basis data terutama terlihat dalam dukungannya terhadap operasi internal perusahaan, termasuk interaksi dengan pelanggan dan pemasok. Basis data juga menjadi penyimpanan bagi informasi administratif dan data yang bersifat lebih khusus, seperti model teknik atau ekonomi. Pengelompokan basis data dapat dilakukan berdasarkan tipe-tipe, model, dan hierarki, dengan model basis data memberikan struktur logis yang menggambarkan hubungan dan aturan dalam penyimpanan, pengaturan, dan manipulasi data.

Model basis data sendiri menjadi fondasi bagi setiap aplikasi basis data yang dibangun. Model ini menentukan struktur logis dari basis data, menjelaskan hubungan antar data, serta aturan yang mengatur cara data dapat disimpan dan dimanipulasi. Dengan demikian, setiap aplikasi basis data akan dikembangkan berdasarkan model data tertentu, menggambarkan landasan yang kokoh bagi manajemen dan pemanfaatan informasi dalam suatu organisasi.

2.1.2 Komponen Basis Data

Komponen basis data mencakup berbagai elemen yang bersatu untuk membentuk sistem basis data yang fungsional. Perangkat keras (hardware) menjadi fondasi utama sebagai media penyimpanan elektronik yang memungkinkan data atau informasi disimpan secara terstruktur. Sistem operasi (operating system) berperan krusial dalam mengatur dan mengontrol seluruh operasi pada sistem basis data, menciptakan lingkungan yang optimal untuk manajemen data.

Basis data (database) sendiri berfungsi sebagai gudang data yang terorganisir, menyediakan tempat penyimpanan yang efisien untuk data yang mudah diperbarui, dikelola, dan diakses. Sistem Pengelolaan Basis Data (DBMS) merupakan komponen yang mengelola segala aspek data, termasuk penyimpanan, pengambilan, dan pengeditan data, memastikan integritas dan keamanan informasi. Pemakai (user) menjadi elemen yang berinteraksi langsung dengan sistem basis data, memiliki wewenang untuk mengakses informasi yang dibutuhkan.

Selain itu, ada pula aplikasi (perangkat lunak) tambahan yang bersifat opsional, berperan dalam menunjang kinerja basis data dan memperkuat perannya sebagai pusat penyimpanan data. Komponen-komponen ini bersinergi untuk menciptakan sistem basis data yang efisien, andal, dan responsif terhadap kebutuhan pemakainya.

2.1.3 Arsitektur Basis Data

Arsitektur basis data, sebagai struktur atau kerangka kerja, memainkan peran sentral dalam mengorganisasi, menyimpan, dan mengelola data secara terpusat. Terdiri dari tiga lapisan utama, arsitektur ini memberikan fondasi yang kokoh untuk sistem basis data. Lapis internal berkaitan dengan aspek penyimpanan fisik data, mendefinisikan cara data disimpan pada media penyimpanan. Lapis konseptual memberikan gambaran tentang struktur data dalam basis data dan hubungan antar data. Sementara itu, lapis eksternal berkaitan dengan tampilan yang terlihat oleh pemakai akhir, memberikan akses yang terfokus pada kebutuhan individu.

Dalam kerangka arsitektur ini, komponen-komponen basis data menjadi unsur kunci dalam mendukung fungsionalitasnya. Perangkat keras sebagai media penyimpanan elektronik, sistem operasi yang mengontrol operasi basis data, dan basis data sebagai gudang data terorganisir adalah bagian integral dari struktur tersebut. Sistem pengelolaan basis data (DBMS) memainkan peran penting dalam mengelola, menyimpan, dan memanipulasi data, sementara pemakai memiliki peran sebagai pengguna akhir yang berinteraksi dengan sistem. Aplikasi tambahan, bersifat opsional namun memperkuat peran sebagai pusat penyimpanan data, menunjukkan kompleksitas dan kelengkapan arsitektur basis data dalam mendukung kebutuhan organisasi modern.

2.2 Entity Relationship Diagram (ERD)

2.2.1 Pengertian ERD

Entity Relationship Diagram (ERD) adalah suatu bentuk diagram yang menjelaskan hubungan antar objek-objek data yang mempunyai hubungan antar relasi. ERD digunakan untuk menyusun struktur data dan hubungan antar data, dan untuk menggambarannya digunakan notasi, simbol, bagan, dan lain sebagainya 12.

ERD terdiri dari beberapa komponen, seperti entitas (entity), atribut (field), dan relasi (relation). Entitas direpresentasikan sebagai kotak dengan atribut-atribut yang terkait dengan entitas tersebut. Atribut adalah karakteristik dari entitas yang didefinisikan oleh entitas itu sendiri. Relasi adalah hubungan antara dua atau lebih entitas. ERD sangat penting dalam pengembangan database karena memudahkan pengembang database untuk memvisualisasikan struktur database dengan jelas dan memahami bagaimana entitas saling terkait.

2.2.2 Kardinalitas Relasi pada ERD

Kardinalitas relasi adalah konsep penting dalam Entity Relationship Diagram (ERD) yang digunakan untuk memetakan hubungan antara entitas dalam basis data. Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain. Dalam ERD, hubungan (relasi) dapat terdiri dari sejumlah entitas yang disebut derajat relasi. Derajat relasi maksimum disebut dengan kardinalitas, sedangkan derajat minimum disebut dengan modalitas. Kardinalitas relasi yang terjadi antara dua himpunan entitas dapat berupa:

1. Satu ke satu (One to one) : Setiap entitas pada himpunan entitas A dapat berelasi dengan paling banyak 1 (satu) entitas pada himpunan entitas B, demikian juga sebaliknya. Contohnya adalah pria menikahi wanita (asumsi tidak ada poligami).
2. Satu ke banyak (One to many) : Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, di mana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak satu entitas pada himpunan entitas A. Contohnya adalah ibu mempunyai anak.

3. Banyak ke banyak (Many to many) : Setiap entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya. Contohnya adalah mahasiswa mengambil mata kuliah.

Kardinalitas relasi sangat penting dalam pengembangan basis data karena memudahkan pengembang database untuk memvisualisasikan hubungan antar entitas dengan jelas dan memahami bagaimana entitas saling terkait. Dalam ERD, kardinalitas relasi ditambahkan pada setiap relasi yang ada untuk mempermudah membaca ERD. Kardinalitas relasi juga membantu pengembang database untuk memastikan bahwa basis data yang dibuat memenuhi kebutuhan pengguna dan dapat diakses dengan mudah.

2.3 Normalisasi Tabel

2.3.1 Bentuk Normal Tahap 1 (1NF)

Tahap 1 Normalisasi (1NF) menjadi langkah krusial dalam proses normalisasi basis data dengan tujuan utama mengurangi redundansi dan memastikan dependensi data yang efisien. Fokus utama pada tahap ini adalah memastikan bahwa setiap kolom dalam tabel memiliki nilai atomik yang tidak terpecah, dan pada saat yang sama tidak ada atribut yang berulang atau berlebihan. Dengan kata lain, setiap sel dalam tabel harus mengandung nilai tunggal, mencegah adanya sekumpulan nilai atau struktur data tertentu yang dapat mengakibatkan kompleksitas dan ketidakefisienan dalam penyimpanan data.

Proses tahap 1NF melibatkan dekomposisi data berdasarkan sampel star table atau universal, di mana tabel tersebut berisi kumpulan data yang masih menunjukkan indikasi relasi redundansi dan atribut multivalued. Ini menjadi langkah dasar yang harus dilalui agar basis data dapat dianggap baik dan efisien. Dengan demikian, tahap ini memberikan dasar yang kuat untuk memastikan bahwa informasi disimpan dengan cara yang optimal, memenuhi standar kebersihan data, dan mendukung kinerja yang optimal dalam sistem basis data.

2.3.2 Bentuk Normal Tahap 2 (2NF)

Bentuk Normal Tahap 2 (2NF) merupakan langkah berikutnya dalam rangkaian normalisasi basis data, yang mengikuti tahap 1NF. Pada tahap ini, setiap tabel harus memenuhi kriteria 1NF dan tidak boleh memiliki atribut yang hanya bergantung pada sebagian dari kunci utama tabel. Dengan kata lain, setiap atribut dalam tabel harus

sepenuhnya tergantung pada kunci utama tabel secara keseluruhan. Jika terdapat atribut yang hanya bergantung pada sebagian dari kunci utama, langkah selanjutnya adalah melakukan dekomposisi atau penguraian tabel menjadi dua atau lebih tabel baru.

Proses dekomposisi tersebut memastikan bahwa setiap tabel baru yang dihasilkan memiliki kunci utama yang unik dan tidak ada atribut yang bergantung pada sebagian dari kunci utama. Tahap 2NF menjadi langkah kritis untuk mencapai basis data yang baik dan efisien, memastikan struktur data yang lebih terorganisir dan mengurangi ancaman terhadap integritas data. Dengan melewati tahap ini, basis data menjadi lebih terstruktur dan siap untuk mengelola informasi dengan lebih efektif sesuai dengan prinsip-prinsip normalisasi.

2.3.3 Bentuk Normal Tahap 3 (3NF)

Tahap 3 Normalisasi (3NF) menandai langkah selanjutnya dalam proses normalisasi basis data setelah 2NF. Pada tahap ini, persyaratan utama adalah bahwa setiap tabel harus memenuhi kriteria 2NF, di mana tidak boleh ada atribut yang bergantung pada atribut non-kunci lain dalam tabel. Dengan kata lain, setiap atribut dalam tabel harus tergantung hanya pada kunci utama tabel dan tidak memiliki dependensi pada atribut non-kunci lain. Jika terdapat atribut yang melanggar kriteria ini, tahap selanjutnya melibatkan dekomposisi atau penguraian tabel menjadi dua atau lebih tabel baru.

Proses dekomposisi tersebut memastikan bahwa setiap tabel baru yang dihasilkan memiliki kunci utama yang unik dan tidak memiliki atribut yang bergantung pada atribut non-kunci lain dalam tabel. Tahap 3NF membawa basis data ke tingkat kebersihan dan efisiensi yang lebih tinggi, mengurangi kemungkinan redundansi dan menjaga integritas data. Dengan melalui tahap ini, basis data menjadi lebih terstruktur dan siap untuk mendukung aplikasi serta kebutuhan bisnis dengan optimal.

2.4 Basis Data MySQL

Basis data MySQL adalah sistem manajemen basis data relasional (RDBMS) yang memiliki ciri khas sebagai solusi open source yang populer. Dikembangkan oleh Oracle Corporation, MySQL menawarkan keberlanjutan, kinerja yang handal, dan kemudahan penggunaan. MySQL menggunakan bahasa query SQL (Structured Query Language) untuk interaksi dengan basis data, memungkinkan pengguna untuk membuat, mengelola, dan mengambil data dengan mudah. Kelebihan MySQL meliputi dukungan integrasi dengan

berbagai bahasa pemrograman, kapasitas RAM yang efisien, kemampuan untuk mendukung pengguna multi, dan keamanan data yang baik. MySQL juga dikenal karena struktur tabelnya yang fleksibel dan dukungan untuk tipe data yang beragam, memberikan keleluasaan bagi pengembang dalam merancang dan mengelola skema basis data yang kompleks.

Meskipun MySQL memiliki banyak kelebihan, beberapa kekurangan juga perlu diperhatikan. Misalnya, MySQL kurang cocok untuk aplikasi game dan mobile, serta menghadapi tantangan dalam mengelola database yang sangat besar. Selain itu, dukungan teknis untuk MySQL bisa menjadi kurang memadai, terutama jika dibandingkan dengan solusi berbayar. Meski demikian, MySQL tetap menjadi pilihan utama banyak organisasi dan pengembang aplikasi berkat keandalannya, skalabilitas, dan dukungan komunitas open source yang kuat.

Bab III

Perancangan Basis Data

3.1 Tinjauan Instansi

3.1.1 Proses Bisnis / Proses Kerja Perusahaan

Proses bisnis di perusahaan BlissBlocks, yang merupakan perusahaan mainan konstruksi, melibatkan serangkaian tahapan utama yang mencerminkan alur kerja dan kegiatan inti perusahaan. Tahapan ini mencakup riset dan pengembangan produk untuk mengidentifikasi tren pasar dan menciptakan produk inovatif. Tim desain kemudian mengambil peran penting dalam mengembangkan konsep dan desain produk yang estetis dan menarik. Proses produksi dan manufaktur dilakukan untuk mengubah desain menjadi produk fisik dengan standar kualitas tinggi.

Setelahnya, produk didistribusikan ke berbagai pengecer dan distributor melalui proses distribusi dan logistik yang efisien. Upaya pemasaran dan promosi dilakukan untuk meningkatkan kesadaran merek dan mempromosikan produk BlissBlocks. Penjualan dan layanan pelanggan menjadi tahap penting dalam berinteraksi dengan konsumen, baik melalui sistem penjualan online maupun offline, dengan layanan pelanggan yang responsif.

Pemeliharaan kualitas dan inovasi berkelanjutan menjadi fokus perusahaan untuk memastikan produk tetap relevan dan memenuhi ekspektasi konsumen. Pengelolaan persediaan yang efisien diterapkan untuk memastikan ketersediaan produk, sementara produksi berkelanjutan dan ramah lingkungan menjadi bagian integral dari proses operasional BlissBlocks.

Detail dari proses-proses ini dapat disesuaikan dengan ukuran perusahaan, model bisnis, dan strategi operasional yang diadopsi oleh BlissBlocks. Sebagai inti dari keberlanjutan dan keberhasilan perusahaan, proses bisnis tersebut terintegrasi untuk mencapai efisiensi dan memberikan nilai tambah bagi konsumen BlissBlocks.

3.2 Perancangan ERD

Sebelum memasuki tahap perancangan Entity-Relationship Diagram (ERD), langkah awal yang krusial dalam pengembangan basis data adalah pemahaman mendalam terhadap data yang akan diwakili. Pada tahap ini, fokus utama adalah menggali informasi yang terdapat pada setiap tabel yang akan diintegrasikan ke dalam ERD. Proses ini melibatkan analisis rinci

terhadap struktur dan atribut-atribut pada setiap tabel, seperti inventaris, bagian (parts), kategori bagian, dan warna. Data understanding ini menjadi landasan penting untuk menghasilkan ERD yang representatif dan mencerminkan kebutuhan serta hubungan antar entitas secara akurat.

Dalam konteks tabel-tabel tersebut, mari tinjau informasi yang dimiliki oleh masing-masing:

- **Tabel Inventories**

Berisi informasi tentang inventaris, dengan kolom-kolom seperti id yang mewakili identifikasi unik, version untuk menandakan versi inventaris, dan set_num yang merepresentasikan nomor set inventaris. Data dalam tabel ini mencakup rincian setiap inventaris yang dimiliki perusahaan, membantu dalam pelacakan dan manajemen stok secara efektif.

- **Tabel Inventory_part**

Menunjukkan hubungan antara inventaris dan bagian (parts) yang dimiliki oleh inventaris tersebut. Kolom inventory_id dan part_num berfungsi sebagai kunci asing yang menghubungkan kedua tabel. Informasi seperti color_id, quantity, dan is_spare memberikan rincian tentang bagian-bagian yang terkait dengan setiap inventaris, termasuk warna, jumlah, dan status keberadaannya sebagai Inventory_part.

- **Tabel Parts**

Berisi informasi rinci tentang bagian-bagian (parts) yang digunakan dalam inventaris. Kolom part_num menjadi kunci utama yang unik, sementara name memberikan deskripsi singkat tentang nama bagian, dan part_cat_id merujuk ke kategori bagian. Tabel ini memungkinkan identifikasi mudah dan manajemen bagian inventaris untuk keperluan perancangan dan pemeliharaan.

- **Tabel Part_categories**

Berperan dalam mengelompokkan berbagai bagian menjadi kategori tertentu. Dengan kolom id sebagai kunci utama dan name sebagai deskripsi kategori, tabel ini membantu dalam mengorganisir dan mengklasifikasikan bagian-bagian inventaris, memudahkan pemahaman dan penanganan berbagai kategori dalam perusahaan.

- **Tabel Color**

Memuat informasi terkait warna yang mungkin digunakan pada bagian-bagian inventaris. Dengan kolom id sebagai kunci utama, name untuk menyimpan nama warna, rgb sebagai representasi kode warna RGB, dan is_trans yang menunjukkan

apakah warna tersebut transparan. Data pada tabel ini mendukung pemilihan warna untuk bagian-bagian inventaris, memberikan keberagaman estetika pada produk.

- **Tabel Sets**

Berisi data terkait set Inventaris. Kolom "set_num" digunakan sebagai identifikasi unik untuk set tersebut, "name" mencerminkan nama set, "year" menyajikan tahun rilis produk, "theme_id" mencakup ID tema yang terhubung dengan tabel Themes, dan "num_parts" menunjukkan total bagian yang dimasukkan dalam set tersebut.

- **Inventory_sets**

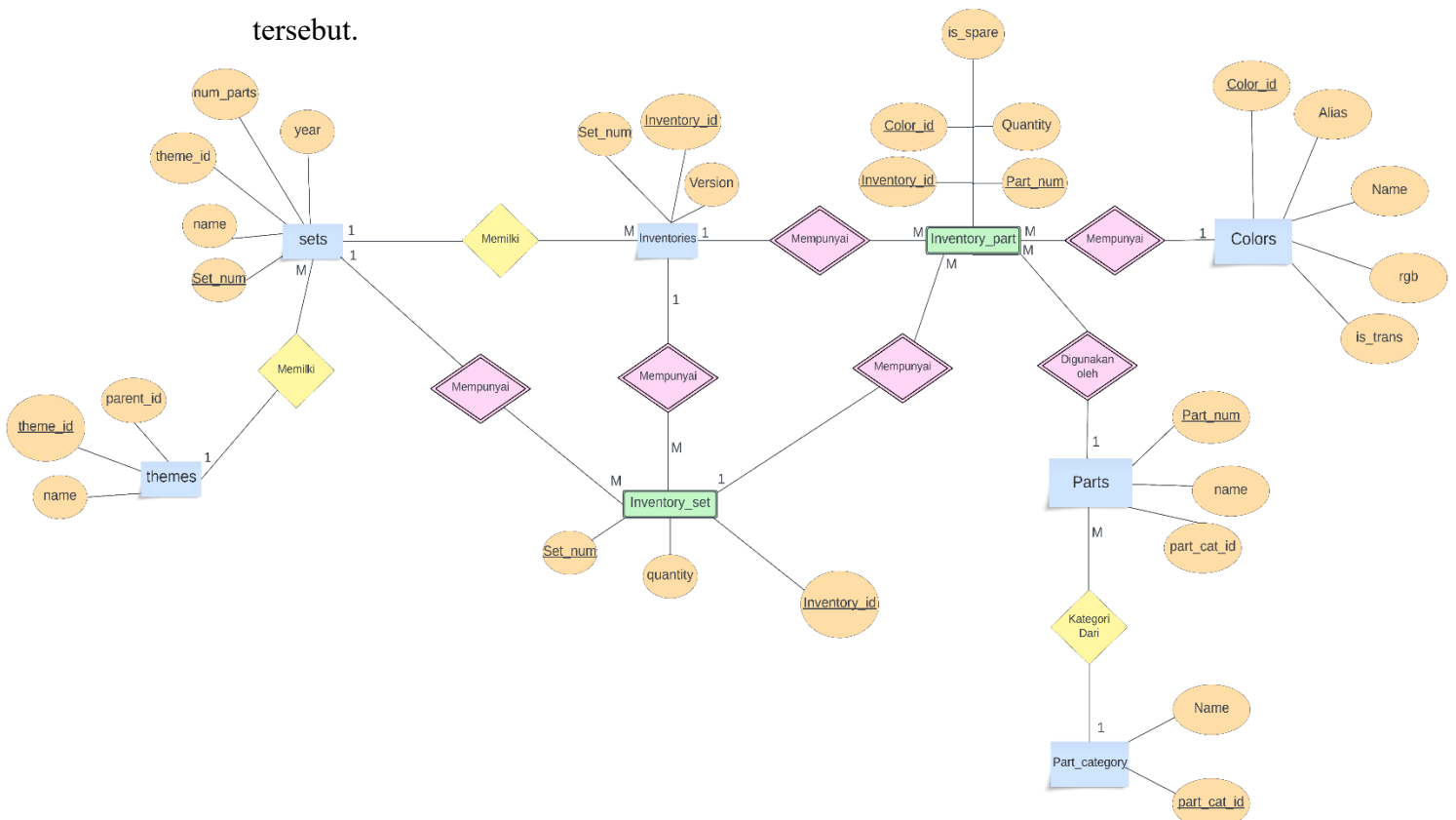
Menyajikan informasi mengenai bagian inventaris yang disertakan dalam set tertentu, termasuk ID inventaris, nomor set, dan jumlah inventaris yang disertakan dalam set tersebut.

- **Themes**

Memuat informasi tentang tema-tema set mainan, dengan kolom ID unik, nama tema, dan ID tema yang lebih besar jika tema tersebut bagian dari tema yang lebih besar.

3.2.1 Entity Relationship Diagram (ERD)

Dengan landasan pemahaman yang mendalam terhadap setiap detail yang tersimpan dalam tabel Inventories, Inventory_part, Parts, Part_categories, dan Color, berikut adalah Entity-Relationship Diagram (ERD) yang telah dirancang berdasarkan informasi tersebut.



Pertama, terdapat relasi lemah antara entitas Inventories dan Inventory_Part yang menunjukkan bahwa satu inventaris dapat terdiri dari banyak bagian (parts). Kedua, entitas Colors berelasi lemah dengan Inventory_Part, dimana satu warna dapat digunakan oleh banyak item inventaris.

Selanjutnya, entitas Parts juga memiliki relasi lemah dengan Inventory_Part, yang menunjukkan bahwa satu jenis part dapat digunakan pada banyak item inventaris yang berbeda.

Terakhir, entitas Part_Categories yang berfungsi untuk mengkategorikan part, memiliki relasi satu-ke-banyak dengan entitas Parts. Relasi ini satu-arah dimana satu kategori part dapat memiliki banyak jenis part dengan spesifikasi yang berbeda-beda di dalamnya.

Selain itu, terdapat empat relasi tambahan yang melibatkan entitas Inventories, Sets, Themes, Inventory_Sets, dan Inventory_Parts. Pertama, antara Inventories dan Inventory_Sets terdapat relasi one-to-many yang menunjukkan bahwa satu inventaris dapat terkait dengan banyak set inventaris.

Kedua, relasi one-to-many juga terjadi antara Sets dan Inventory_Sets, mengindikasikan bahwa satu set dapat memiliki banyak keterkaitan dengan inventory_sets. Ketiga, Themes memiliki relasi one-to-many dengan Sets. Kemudian juga ada hubungan sets dengan inventories dengan hubungan one to many, dimana satu set dapat terhubung dengan banyak inventaris. Terakhir, antara Inventory_Sets dan Inventory_Parts terdapat relasi one-to-many.

Secara keseluruhan, terdapat 9 relasi pada ERD inventaris ini, 6 diantaranya adalah relasi lemah dan 3 relasi kuat. Relasi-relasi ini menggambarkan interaksi dinamis dan saling ketergantungan antar entitas dalam basis data inventaris.

3.2.2 Implementasi ERD ke Basis Data Fisik

Setelah merancang Entity Relationship Diagram (ERD), langkah berikutnya adalah mentransformasikan struktur konseptual tersebut ke dalam bentuk basis data fisik. Proses ini melibatkan beberapa aturan yang perlu diikuti untuk memastikan implementasi yang efisien dan konsisten. Berikut adalah beberapa aturan yang perlu diperhatikan:

1. Aturan 1 :
 - Setiap himpunan entitas (Entitas Kuat) diimplementasikan sebagai tabel (file data).

- Tentukan satu atribut unik sebagai Primary Key.
2. Aturan ke-2 :
 - Relasi 1:1 direpresentasikan dengan menambahkan atribut-atribut relasi ke salah satu tabel.
 - Pilihan peleburan relasi ke salah satu tabel ditentukan berdasarkan Derajat Relasi Minimum.
 - Jika Derajat Relasi Minimum sama, peleburan relasi dilakukan ke tabel dengan jumlah row data lebih sedikit.
 3. Aturan ke-3 :
 - Relasi 1:M direpresentasikan dengan menambahkan atribut key dari entitas-I ke tabel yang mewakili entitas-II.
 - Atribut key dari entitas-I menjadi atribut tambahan bagi entitas-II.
 4. Aturan ke-4 :
 - Relasi M:M direpresentasikan dalam bentuk tabel khusus dengan foreign key dari key-key entitas yang dihubungkannya
 5. Aturan ke-5
 - Untuk setiap himpunan entitas lemah, buat tabel baru dengan foreign key yang diambil dari primary key entitas kuat.
 - Primary key terbentuk dari gabungan primary key entitas kuat dan partial key entitas lemah (jika ada).
 6. Aturan ke-6 :
 - Hanya atribut sederhana dari atribut komposit yang disertakan dalam tabel.
 - Pilih salah satu key attribute dari entitas kuat sebagai Primary Key. Pilih salah satu key attribute dari entitas kuat sebagai Primary Key.
 7. Aturan ke-7
 - Untuk setiap multivalued attribute, buat tabel baru dan tambahkan foreign key sebagai Primary Key dari entitas yang memiliki attribute tersebut.

Dari ke-7 aturan di atas, maka terdapat langkah-langkah yang harus dilakukan untuk memulai membuat tabel bentuk fisik :

1. Step 1:

- Pahami semua entitas regular/strong dari diagram tersebut dan kemudian buat relasi (tabel) yang sesuai yang mencakup semua atribut sederhana. Pilih salah satu atribut sebagai kunci utama.
- Jika bersifat komposit, gabungan atribut sederhana membentuk kunci utama.
- Untuk ER-Diagram yang diberikan, kita memiliki Inventories, Colors, Parts, themes, sets, dan Part_category sebagai entitas strong/regular, karena mereka dikelilingi oleh satu persegi panjang.

Jadi, kita membuat relasi yang sesuai seperti yang digambarkan dalam gambar di bawah ini

Part_category

<u>Part_cat_id</u>	Name

themes

<u>theme_id</u>	name	parent_id

Inventories

<u>Inventory_id</u>	Version	Set_num

Parts

<u>Part_num</u>	Name	Part_cat_id

colors

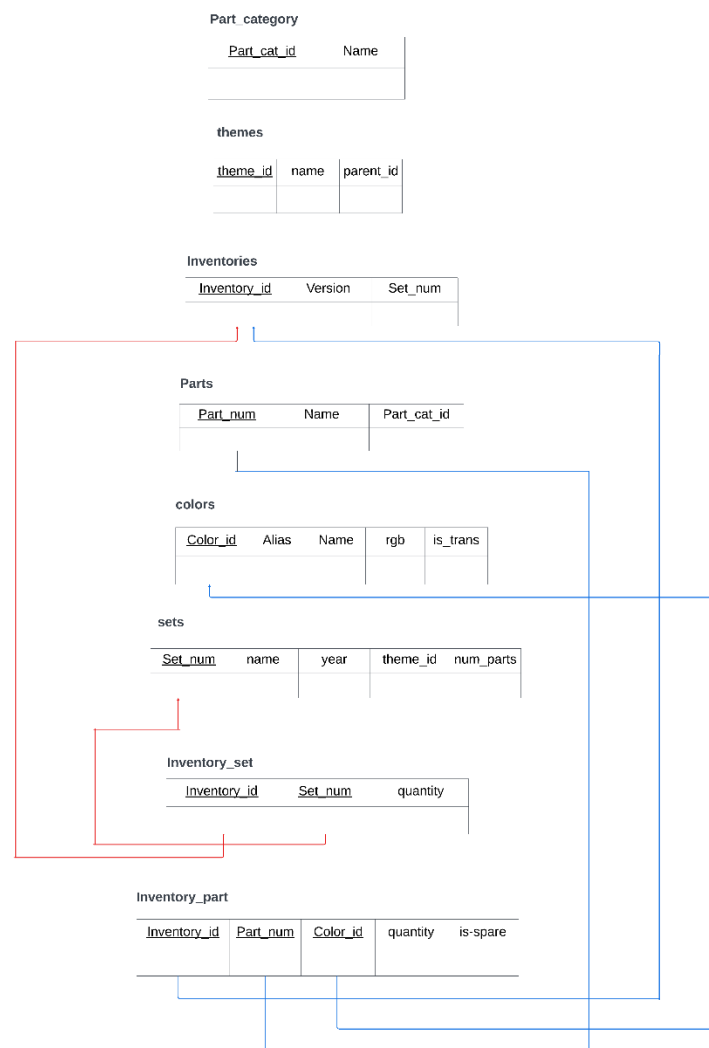
<u>Color_id</u>	Alias	Name	rgb	is_trans

sets

<u>Set_num</u>	name	year	theme_id	num_parts

2. Step 2 :

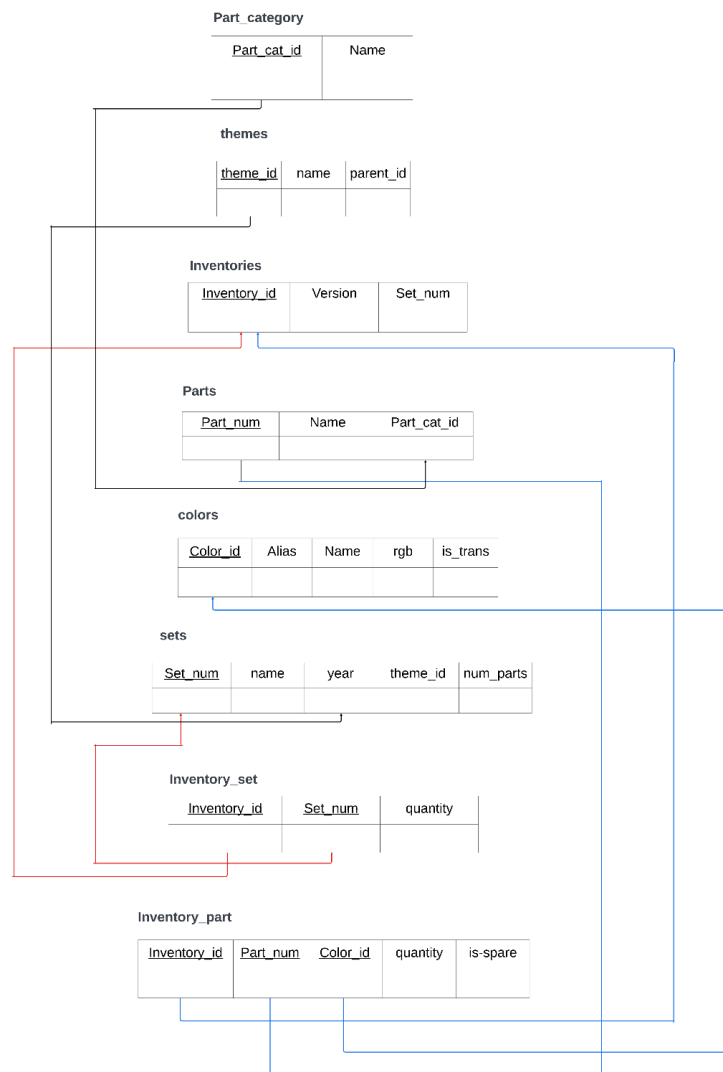
- Identifikasi jenis entitas lemah dari diagram dan buat relasi (tabel) yang sesuai yang mencakup semua atribut sederhana.
- Tambahkan sebagai kunci asing semua atribut kunci utama pada entitas yang merupakan pemilik (owner entity).
- Kunci utama merupakan kombinasi dari semua atribut kunci utama dari pemilik dan kunci utama dari entitas lemah tersebut.
- Pada ER-Diagram yang diberikan, kita memiliki Inventory_part dan Inventory_sets sebagai entitas lemah, karena terdapat dalam suatu persegi panjang ganda yang menunjukkan bahwa entitas tersebut lemah.
- Jadi, kita membuat relasi (tabel) Dependent yang ditunjukkan dalam gambar di bawah ini.



3. Step 3 :

- Entitas yang memiliki relasi 1-ke-N ini, pilih relasi sebagai S untuk jenis pada sisi N dari relasi dan yang lainnya sebagai T. Kemudian tambahkan sebagai kunci asing ke S semua atribut kunci utama dari T.
- Pada diagram ER yang diberikan, terdapat dua relasi 1-ke-N antara entitas Part_category – Parts dan themes – sets .

Pemetaan 1 ke-N antara Part_category – Parts ditunjukkan dalam gambar di bawah ini.



Dengan tidak adanya relasi lain seperti one-to-one atau many-to-many, dan tanpa keberadaan atribut multivalued, proses pemetaan basis data fisik telah selesai.

3.3 Normalisasi Tabel

Setelah mengubah ERD ke bentuk basis data fisik, langkah selanjutnya yaitu melakukan normalisasi. Normalisasi tabel adalah proses desain basis data untuk mengurangi redundansi data dan memastikan integritas data. Proses normalisasi terdiri dari beberapa tingkatan, yaitu 1NF (First Normal Form), 2NF (Second Normal Form), dan 3NF (Third Normal Form). Berikut adalah aturan untuk setiap tingkat normalisasi:

1. 1NF (First Normal Form)
 - Nilai Atomik
 - Setiap sel di dalam tabel harus berisi nilai atomik, artinya nilainya tidak dapat dipecah lagi.
 - Struktur tabel yang teratur
 - Setiap kolom harus memiliki nama yang unik.
 - Urutan kolom tidak memiliki arti khusus.
2. 2NF (Second Normal Form)
 - Sudah dalam 1NF
 - Tidak ada ketergantungan parsial
 - Setiap non-kunci harus sepenuhnya bergantung pada seluruh kunci primer.
3. 3NF (Third Normal Form)
 - Sudah dalam 2 NF
 - Tidak ada ketergantungan transitif
 - Setiap kolom non-kunci harus tidak bergantung pada kolom non-kunci lainnya (transitif), kecuali jika ketergantungan tersebut adalah terhadap kunci primer.

3.3.1 Bentuk Normal Tahap 1 (1NF)

Setelah mengetahui hal-hal yang harus dilakukan untuk mencapai 1NF, langkah selanjutnya adalah menganalisis setiap tabel secara individual. Untuk datanya sendiri dapat dilihat pada link berikut <https://docs.google.com/spreadsheets/d/18tq-l-bJLSTveNGpQGD936CKPcjvZuY5S02NFpxrr7M/edit?usp=sharing>. Berikut preview dari 5 data teratas untuk masing masing tabel.

Part_category

id	name
1	Baseplates
2	Bricks Printed
3	Bricks Sloped
4	Duplo, Quatro and Primo
5	Bricks Special
6	Bricks Wedged

Inventories

id	version	set_num
1	1	7922-1
3	1	3931-1
4	1	6942-1
15	1	5158-1
16	1	903-1

Parts

part_num	name	part_cat_id
11816pr0005	Minifig Head Modified - Friends - Green Eyes, Pink Lips and Closed Mouth Print	13
12708pr01	Hemisphere [Bauble Half] with White Dots Print	20
12825	Tile Special 1 x 1 with Clip with Rounded Tips	15
15573	Plate Special 1 x 2 with 1 Stud with Groove and Inside Stud Holder (Jumper)	9
2343	Minifig Goblet	27

Colors

id	Alias	name	rgb	is_trans
-1	UK	Unknown	0033B2	f
0	BL	Black	05131D	f
1	BU	Blue	0055BF	f
2	GR	Green	237841	f
3	DT	Dark Turquoise	008F9B	f

Inventory_part

inventory_id	part_num	color_id	quantity	is_spare
1	48379c01	72	1	f
1	48395	7	1	f
1	mcsport6	25	1	f
1	paddle	0	1	f
3	11816pr0005	78	1	f

Sets

set_num	name	year	theme_id	num_parts
10008-1	2 x 4 Roof Tile Sand Red	2001	254	50
10071-1	Mr. Bunny	2003	229	25
10128-1	Train Level Crossing	2003	239	326
10189-1	Taj Mahal	2008	276	5922

Themes

set_num	name	year	theme_id	num_parts
10008-1	2 x 4 Roof Tile Sand Red	2001	254	50
10071-1	Mr. Bunny	2003	229	25
10128-1	Train Level Crossing	2003	239	326
10189-1	Taj Mahal	2008	276	5922
1076-3	Advent Calendar 1999 (Day 2) Snowman	1999	217	11

Inventory_sets

inventory_id	set_num	quantity
35	75911-1	1
35	75912-1	1
39	75048-1	1
39	75053-1	1
50	4515-1	1

Setelah dilakukan penelusuran pada seluruh tabel, terlihat bahwa setiap tabel telah memenuhi persyaratan 1 Normalisasi (1NF). Setiap baris dalam tabel memiliki nilai atomik untuk setiap kolom, dan tidak ada nilai yang dapat dibagi lagi. Selain itu, tidak ada data yang kosong pada seluruh tabel. Oleh karena itu, tidak diperlukan perubahan lebih lanjut untuk mencapai 1NF pada data ini. Semua data sudah sesuai dengan persyaratan 1NF dan tidak terdapat nilai yang kosong.

3.3.2 Bentuk Normal Tahap 2 (2NF)

Setelah berhasil menerapkan 1NF, langkah selanjutnya adalah menuju normalisasi ke bentuk 2NF. Dengan hasil analisis yang dilakukan, ternyata tabel-tabel di atas sudah memenuhi persyaratan 2NF, di mana setiap atribut non-kunci memiliki ketergantungan penuh pada kunci utama. Secara spesifik, pada tabel Part_category,

kolom name tergantung pada id; pada tabel Inventories, kolom version dan set_num tergantung pada id. Selanjutnya, pada tabel Parts, kolom name dan part_cat_id tergantung pada part_num; pada tabel Colors, kolom alias, name, rgb, dan is_trans tergantung pada id; dan terakhir, pada tabel Inventory_part, kolom quantity dan is_spare tergantung pada kombinasi inventory_id, part_num, dan color_id.

Maka, tabel-tabel di atas masih berada dalam bentuk yang sama seperti yang terakhir kali dalam 1NF.

Part_category

id	name
1	Baseplates
2	Bricks Printed
3	Bricks Sloped
4	Duplo, Quatro and Primo
5	Bricks Special
6	Bricks Wedged

Inventories

id	version	set_num
1	1	7922-1
3	1	3931-1
4	1	6942-1
15	1	5158-1
16	1	903-1

Parts

part_num	name	part_cat_id
11816pr0005	Minifig Head Modified - Friends - Green Eyes, Pink Lips and Closed Mouth Print	13
12708pr01	Hemisphere [Bauble Half] with White Dots Print	20
12825	Tile Special 1 x 1 with Clip with Rounded Tips	15
15573	Plate Special 1 x 2 with 1 Stud with Groove and Inside Stud Holder (Jumper)	9
2343	Minifig Goblet	27

Colors

id	Alias	name	rgb	is_trans
-1	UK	Unknown	0033B2	f
0	BL	Black	05131D	f
1	BU	Blue	0055BF	f
2	GR	Green	237841	f
3	DT	Dark Turquoise	008F9B	f

Inventory_part

inventory_id	part_num	color_id	quantity	is_spare
1	48379c01	72	1	f
1	48395	7	1	f
1	mcsport6	25	1	f
1	paddle	0	1	f
3	11816pr0005	78	1	f

Sets

set_num	name	year	theme_id	num_parts
10008-1	2 x 4 Roof Tile Sand Red	2001	254	50
10071-1	Mr. Bunny	2003	229	25
10128-1	Train Level Crossing	2003	239	326
10189-1	Taj Mahal	2008	276	5922

Themes

set_num	name	year	theme_id	num_parts
10008-1	2 x 4 Roof Tile Sand Red	2001	254	50
10071-1	Mr. Bunny	2003	229	25
10128-1	Train Level Crossing	2003	239	326
10189-1	Taj Mahal	2008	276	5922
1076-3	Advent Calendar 1999 (Day 2) Snowman	1999	217	11

Inventory_sets

inventory_id	set_num	quantity
35	75911-1	1
35	75912-1	1
39	75048-1	1
39	75053-1	1
50	4515-1	1

3.3.3 Bentuk Normal Tahap 3 (3NF)

Setelah melalui tahap normalisasi 2NF, langkah berikutnya adalah menuju normalisasi tahap 3 (3NF). Tujuan dari normalisasi 3NF adalah memastikan bahwa tidak ada dependensi transitif antara atribut non-kunci. Pada tabel colors terdapat 5 kolom yaitu id, Alias, name, rgb, dan is_trans. Yang dimana id => Alias, name, rgb, is_trans dan Alias => name. Disini terdapat masalah yang dimana name bergantung kepada Alias yang merupakan non-kunci, sehingga diperlukan dekomposisi agar memenuhi kriteria 3NF. Dekomposisi yang dilakukan agar tabel colors memenuhi 3NF yaitu dengan cara membagi tabel colors menjadi 2 tabel, yang pertama tabel berisi kolom id, Alias, rgb, is_trans dan tabel yang kedua yaitu tabel yang berisi kolom Alias dan name.

Tabel colors setelah dinormalisasi bentuk 3 NF

Colors_1

id	Alias	rgb	is_trans
-1	UK	0033B2	f
0	BL	05131D	f
1	BU	0055BF	f
2	GR	237841	f
3	DT	008F9B	f

Colors_2

Alias	name
UK	Unknown
BL	Black
BU	Blue
GR	Green
DT	Dark Turquoise

Untuk tabel-tabel lainnya sudah memenuhi bentuk normal tahap 3 karena tidak ada dependensi transitif antara atribut non-kunci.

Bab IV

Implementasi

4.1 Implementasi ke Database MySQL

Bab ini memfokuskan langkah-langkah implementasi solusi yang telah dirancang ke dalam lingkungan database MySQL. Salah satu aspek utama yang diberikan perhatian adalah struktur tabel, yang menjadi landasan penting dalam menyimpan dan mengelola data secara terstruktur. Dalam bab ini, setiap entitas sistem direpresentasikan sebagai tabel dalam basis data, dan detailnya termasuk atribut-atribut, kunci utama, serta hubungan antar tabel. Pendekatan ini membentuk dasar yang kokoh untuk integrasi data, memastikan keefisienan dan konsistensi dalam penyimpanan informasi yang diperlukan untuk menjalankan solusi yang telah dirancang.

4.1.1 Struktur Tabel

Struktur tabel merupakan elemen kunci dalam perancangan dan implementasi basis data MySQL untuk proyek ini. Pada bagian ini, struktur dari setiap tabel yang telah dirancang, termasuk atribut-atributnya, primary key, dan ketergantungan antar tabel akan dijelaskan secara terperinci. Pemahaman mendalam mengenai struktur tabel sangat penting untuk memastikan integritas data dan efisiensi operasional saat menyimpan, memperbarui, ataupun mengambil informasi.

Berdasarkan hasil normalisasi tabel pada bab sebelumnya, berikut adalah rincian struktur tabel beserta elemen-elemennya:

1. Tabel Part_category

```
CREATE TABLE Part_Category (  
  id INT PRIMARY KEY,  
  name VARCHAR(255) NOT NULL  
);
```

2. Tabel Inventories

```
CREATE TABLE Inventories (  
  id INT PRIMARY KEY,  
  version VARCHAR(255) NOT NULL,  
  set_num VARCHAR(255) NOT NULL  
);
```

3. Tabel Parts

```
CREATE TABLE Parts (  
  part_num VARCHAR(255) PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  part_cat_id INT,  
  FOREIGN KEY (part_cat_id) REFERENCES Part_Category(id)  
);
```

4. Tabel Colors_1

```
CREATE TABLE colors_1 (  
  id INT PRIMARY KEY,  
  alias VARCHAR(255) NOT NULL,  
  name VARCHAR(255) NOT NULL,  
  rgb VARCHAR(6) NOT NULL,  
  is_trans CHAR(1) NOT NULL,  
  INDEX idx_alias (alias) -- Menambahkan indeks pada kolom 'alias'  
);
```

5. Tabel Colors_2

```
CREATE TABLE Colors_2 (  
  alias VARCHAR(255) PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  FOREIGN KEY (alias) REFERENCES Colors_1 (alias)  
);
```

6. Tabel Inventory_Part

```
CREATE TABLE Inventory_Part (  
  inventory_id INT,  
  part_num VARCHAR(255),  
  color_id INT,  
  quantity INT,  
  is_spare CHAR(1) NOT NULL,  
  PRIMARY KEY (inventory_id, part_num, color_id),  
  FOREIGN KEY (inventory_id) REFERENCES Inventory_Sets(inventory_id),  
  FOREIGN KEY (inventory_id) REFERENCES Inventories(id),  
  FOREIGN KEY (part_num) REFERENCES Parts(part_num),  
  FOREIGN KEY (color_id) REFERENCES colors_1(id)
```

7. Tabel themes

```
CREATE TABLE Themes (  
  id INT PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  parent_id INT,  
  FOREIGN KEY (parent_id) REFERENCES Themes(id)  
);
```

8. Tabel sets

```
CREATE TABLE Sets (  
  set_num VARCHAR(50) PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  year INT,  
  theme_id INT,  
  num_parts INT,  
  FOREIGN KEY (theme_id) REFERENCES Themes(id)  
);
```

9. Inventory_sets

```
CREATE TABLE Inventory_Sets (  
  inventory_id INT,  
  set_num VARCHAR(50),  
  quantity INT,  
  PRIMARY KEY (inventory_id, set_num),  
  FOREIGN KEY (inventory_id) REFERENCES Inventories(id),  
  FOREIGN KEY (set_num) REFERENCES Sets(set_num)  
);
```

Desain database ini memiliki sejumlah tabel yang saling berhubungan untuk mengkategorikan, mendefinisikan, dan menyimpan data mengenai parts, warna, set Lego, serta inventories. Tabel Part_Category pada database ini berfungsi untuk mendefinisikan kategori part Lego. Primary key pada Tabel Part_Category adalah kolom id. Selanjutnya tabel Parts menyimpan informasi lengkap terkait masing-masing part Lego, dengan part_num sebagai primary key. Kolom part_cat_id merupakan foreign key pada tabel Parts yang mereferensi kolom id di tabel Part_Category, sehingga memungkinkan setiap part Lego untuk diklasifikasikan ke dalam kategori tertentu.

Tabel colors_1 digunakan untuk mendefinisikan warna Lego berdasarkan nilai RGB dan identitas warna lainnya. Setiap warna diidentifikasi dengan kolom id sebagai primary key. Sementara itu tabel colors_2 merupakan tabel alias dari warna-warna yang telah didefinisikan di tabel colors_1. Kolom alias pada tabel colors_2 berperan sebagai primary key sekaligus foreign key yang mereferensi kolom alias di tabel colors_1. Dengan cara ini setiap alias warna dapat ditelusuri warna aslinya.

Pada tabel Themes, kategori tema Lego set didefinisikan, dengan kolom id sebagai primary key. Informasi rinci tentang set Lego yang diproduksi disimpan di tabel Sets, yang memiliki primary key berupa kolom set_num. Kolom theme_id pada tabel Sets merupakan foreign key yang mereferensi kolom id di tabel Themes, sehingga dapat ditentukan tema dari set Lego tertentu.

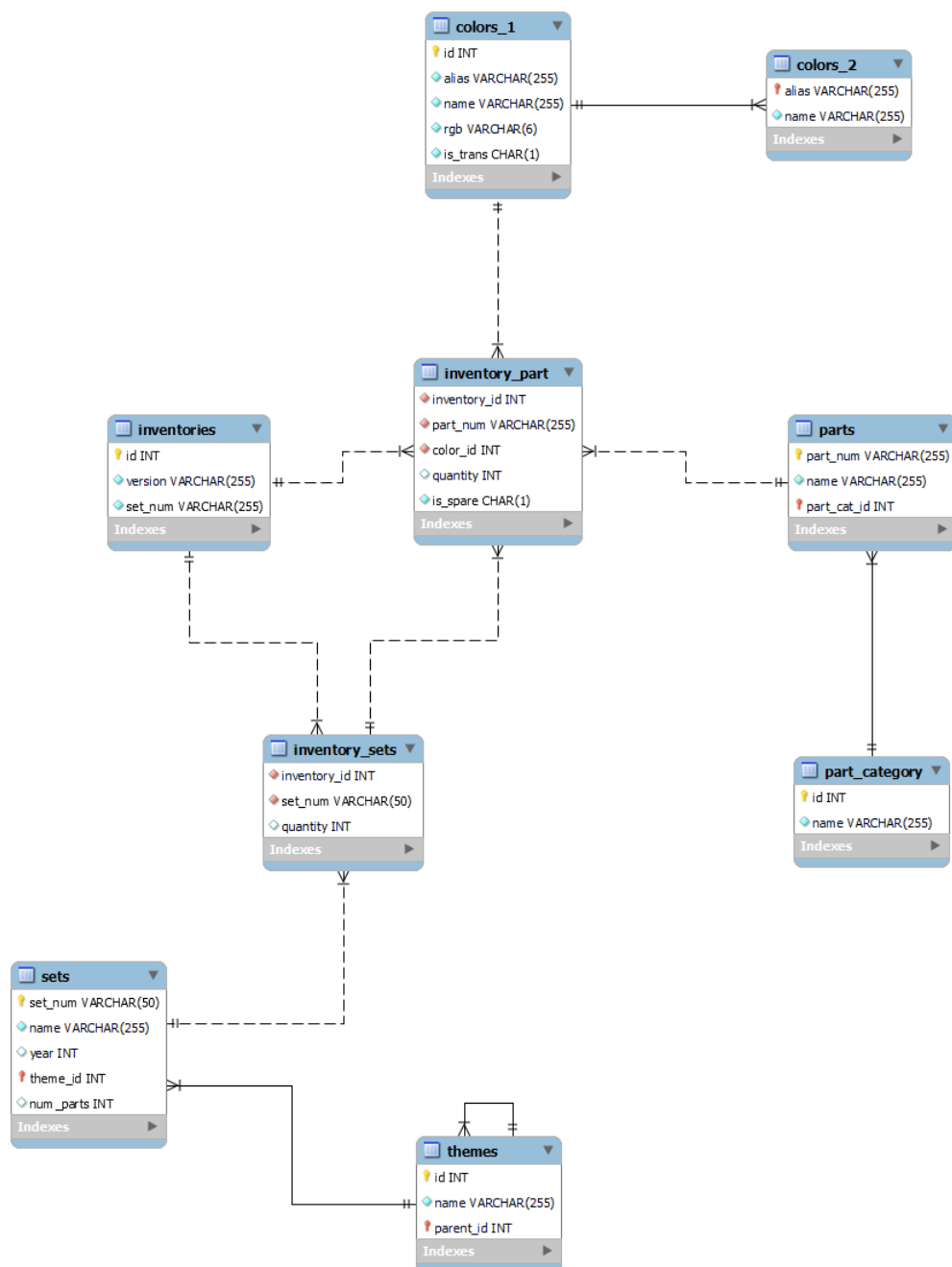
Relasi antara data inventories dengan data set Lego dijembatani oleh tabel Inventory_Sets. Primary key yang digunakan adalah kombinasi inventory_id dan set_num. Di mana inventory_id mereferensi kolom id di tabel Inventories dan set_num mereferensi kolom set_num di tabel Sets.

Terakhir, tabel Inventory_Part mencatat dan mengelola informasi parts yang terdapat pada masing-masing inventory. Kolom inventory_id, part_num, dan color_id digunakan sebagai primary key, yang mereferensi tabel Inventory_Sets, Parts, dan colors_1. Melalui foreign key pada tabel Inventory_Part ini, data parts tersimpan secara terstruktur dan terintegrasi pada level set Lego ataupun keseluruhan inventories.

4.1.1 Relasi Tabel

Dalam bagian ini, akan diuraikan secara rinci struktur tabel yang digunakan dalam implementasi proyek ke database MySQL. Setiap tabel memiliki peran dan hubungan tertentu dalam menyimpan informasi yang diperlukan, dan pemahaman mendalam terhadap struktur tabel menjadi kunci dalam mencapai integritas data yang optimal.

Berikut adalah relasi tabel yang telah diimplementasikan dalam MySQL, sesuai dengan struktur tabel yang telah didefinisikan sebelumnya.



Dalam desain database ini, terdapat beberapa relasi yang perlu diperhatikan untuk memahami keterkaitan antar tabel. Pertama, relasi antara tabel Parts dan Part_Category adalah one-to-many, yang berarti satu kategori (Part_Category) dapat memiliki banyak (Parts). Keterkaitan ini diperlihatkan melalui atribut part_cat_id di tabel Parts yang berfungsi sebagai foreign key, menghubungkan ke primary key (id) di tabel Part_Category.

Selanjutnya, terdapat relasi one-to-one antara tabel Colors_2 dan Colors_1, yang berarti setiap alias hanya terhubung dengan satu nama dan sebaliknya. Keterkaitan ini diwujudkan melalui foreign key alias di tabel Colors_2 yang merujuk pada alias di tabel Colors_1. Oleh karena itu, setiap alias pada tabel Colors_2 memiliki keterkaitan langsung dengan satu dan hanya satu nama pada tabel Colors_1. Dalam konteks ini, hubungan ini dianggap kuat dan ditunjukkan oleh garis yang tidak putus.

Kemudian, relasi one-to-many tampak pada tabel Inventory_Part dengan tabel Inventories, Parts, dan Colors_1. Meskipun terdapat keterkaitan, namun hubungan ini bersifat lemah, ditandai oleh garis yang putus. Kombinasi foreign keys (inventory_id, part_num, color_id) di tabel Inventory_Part merujuk pada kombinasi primary keys di tabel Inventories, Parts, dan Colors_1. Namun, sifat lemah dapat disebabkan karena Inventory_Part merupakan tabel transit yang tidak memiliki primary key sendiri.

Selain itu, pada diagram terlihat bahwa terdapat relasi yang bersifat lemah antara tabel Inventories dengan Inventory_Sets, serta antara tabel Sets dengan Inventory_Sets. Kedua relasi ini ditandai dengan garis putus-putus, yang mengindikasikan sifatnya yang lemah. Inventory_Sets berperan sebagai tabel transit yang menghubungkan data inventories dan sets. Oleh karena itu, Inventory_Sets tidak memiliki primary key sendiri melainkan foreign keys dari kedua tabel yang dihubungkannya.

Di sisi lain, relasi antara tabel Themes dan Sets adalah one-to-many yang bersifat kuat, yang berarti satu tema (Themes) dapat digunakan oleh banyak sets. Foreign key theme_id pada tabel Sets menunjukkan tema dari set tersebut. Selain itu terjadi pula self join pada tabel Themes, dimana kolom parent_id menjadi foreign key yang menghubungkan record tabel itu sendiri. Relasi ini digunakan untuk membentuk hierarki dan kategorisasi tema. Kedua relasi one-to-many ini digambarkan dengan garis tidak putus, menunjukkan hubungan yang kuat.

Berdasarkan penjelasan di atas, dapat disimpulkan bahwa desain database pada gambar tersebut telah memodelkan relasi antar entitas dengan baik. Relasi one-to-many dan one-to-one digambarkan secara akurat melalui penggunaan foreign keys dan primary keys pada setiap tabel, sehingga data dapat tersimpan dan terhubung dengan benar sesuai skema yang dirancang.

Bab V

Penutup

5.1 Kesimpulan

Dalam perjalanan perancangan dan implementasi basis data ini, telah dilakukan serangkaian langkah krusial untuk memastikan keefektifan dan konsistensi penyimpanan serta pengelolaan data. Proses dimulai dengan analisis kebutuhan data untuk menentukan entitas-entitas utama dan hubungan antar entitas tersebut. Skema konseptual dihasilkan berdasarkan analisis ini dan kemudian dinormalisasi hingga mencapai bentuk terbaik guna mengurangi redundansi dan meningkatkan efisiensi.

Melibatkan konsep normalisasi, memastikan setiap tabel dalam basis data ini memenuhi ketentuan Normalisasi hingga bentuk 3NF. Proses ini membantu pemahaman mendalam tentang ketergantungan antar atribut, yang pada gilirannya meningkatkan integritas data dan mengoptimalkan performa query.

Implementasi ke database MySQL menjadi langkah penting berikutnya, di mana struktur tabel mentransformasi skema konseptual menjadi format yang sesuai dengan prinsip-prinsip relational database management system (RDBMS). Dalam tahap ini, relasi antar tabel dibangun dengan memanfaatkan kunci asing (foreign keys) guna menjaga konsistensi data.

Melalui keseluruhan proses ini, telah tercapai keselarasan antara model konseptual dan implementasi nyata dalam database. Hal ini menegaskan pentingnya pemahaman mendalam terhadap kebutuhan dan struktur data untuk menciptakan basis data yang andal, efisien, dan mudah dikelola. Dengan demikian, basis data ini dapat memberikan dukungan yang kokoh bagi aplikasi atau sistem yang menggunakannya.

5.2 Saran

Untuk meningkatkan performa guna mendukung layanan sistem informasi, disarankan melakukan optimasi berkelanjutan terhadap rancangan database. Hal ini dapat dilakukan melalui normalisasi tabel secara berkala, pengelolaan foreign keys secara konsisten, monitoring dan penyesuaian kinerja database, serta penerapan praktik terbaik terkait keamanan data, pemulihan bencana, dan aspek keandalan lainnya. Dengan langkah-langkah perbaikan dan penyempurnaan database secara menyeluruh dan berkesinambungan tersebut, diharapkan sistem dapat terus memberikan layanan prima dan mendukung operasional organisasi.

Daftar Pustaka

- 1NF dan 2NF dan 3NF apakah mereka sama?* (2023, Desember 4). Retrieved from sridianti: <https://www.sridianti.com/gaya-hidup/1nf-dan-2nf-dan-3nf.html>
- Adani, M. R. (2020, Agustus 15). *Mengenal MySQL: Pengertian, Fungsi, serta Kelebihannya*. Retrieved from sekawanmedia: <https://www.sekawanmedia.co.id/blog/pengertian-mysql/>
- Apa itu Basis Data*. (n.d.). Retrieved from aws.amazon: <https://aws.amazon.com/id/what-is/database/>
- Apa itu ERD? Fungsi, Simbol, dan Toolsnya*. (2020, Februari 10). Retrieved from jagoanhosting: <https://www.jagoanhosting.com/blog/erd-apa-sih-itu/>
- Arsitektur Basis Data: Konsep Dasar untuk Mengelola Data*. (2023, Mei 23). Retrieved from ilyasweb: <https://www.ilyasweb.com/arsitektur-basis-data>
- Basis Data (Database): Pengertian, Tujuan, Operasi, dan Bahasa Basis Data*. (n.d.). Retrieved from kmtech.id: <https://www.kmtech.id/post/basis-data-database-pengertian-tujuan-operasi-dan-bahasa-basis-data>
- Basis Data : Pengertian, Komponen, Hirarki, Penggunaan dan Tujuan*. (n.d.). Retrieved from gurupendidikan: <https://www.gurupendidikan.co.id/basis-data/>
- BASIS DATA: Pengertian, Komponen dan Sistem Basis Data (Database)*. (n.d.). Retrieved from carakami: <https://carakami.com/basis-data/>
- Belajar MySQL Dasar : Tutorial MySQL untuk Pemula*. (n.d.). Retrieved from ngodingdata: <https://ngodingdata.com/tutorial-mysql-untuk-pemula/>
- Contoh Script Database MySQL Untuk Pemula*. (2023, Maret 31). Retrieved from BRO.WEB>ID: <https://www.bro.web.id/2023/03/contoh-script-database-mysql.html>
- ISMAIL, R. R. (2012, November 8). *Pengertian Normalisasi pada Database Management System*. Retrieved from https://mahasiswa.ung.ac.id/921409222/home/2012/11/8/normalisasi_database_relasi.html
- K, Y. (2022, April 24). *Pengertian MySQL, Fungsi, dan Cara Kerjanya (Lengkap)*. Retrieved from niagahoster: <https://www.niagahoster.co.id/blog/mysql-adalah/>
- M.Ramadhan. (2021, Oktober 3). *Normalisasi Basis Data: 1NF, 2NF, 3NF*. Retrieved from medium: <https://medium.com/telematika/normalisasi-1nf-2nf-3nf-1018bdecf028>
- Meilinaeka. (2023, Maret 31). *Tahapan Dalam Proses Normalisasi Data dan Contoh Soal*. Retrieved from meilinaeka's blog: <https://meilinaeka.staff.telkomuniversity.ac.id/2023/03/31/tahapan-dalam-proses-normalisasi-data-dan-contoh-soal/>
- N, S. (2023, Agustus 15). *Contoh Normalisasi Database Mahasiswa: Mengoptimalkan Efisiensi dan Integritas Data*. Retrieved from PemburuKode: <https://pemburukode.com/contoh-normalisasi-database-mahasiswa/>

- Pengertian Database, Terminologi & Pemodelannya.* (n.d.). Retrieved from serupa.id: <https://serupa.id/pengertian-basis-data-database-terminologi-pemodelannya/>
- Pengertian Komponen Basis Data & Perangkat Pendukungnya.* (2021, July 25). Retrieved from qwords: <https://qwords.com/blog/komponen-basis-data/>
- Proses Normalisasi Tabel Database.* (2010, Februari 17). Retrieved from hilman: <https://hilman.web.id/blog/808/proses-normalisasi-tabel-database.html>
- Rahmawati, R. (n.d.). *Normalisasi Database: Pengertian, Tahapan dan Contoh.* Retrieved from DosenIT.com: <https://dosenit.com/software/normalisasi-database>
- Share, B. (2020, November 23). *Normalisasi Basis Data Bentuk Normal Pertama (1NF).* Retrieved from bakhel: <https://www.bakhel.com/2020/11/normalisasi-basis-data-bentuk-normal-pertama-1NF.html>
- Sulthon, A. (2023, Mei 27). *Cara membuat ERD: Simbol, Entitas, Atribut Termudah.* Retrieved from Domainsia: <https://www.domainsia.com/berita/pengertian-erd-adalah/>
- Thabroni, G. (2023, November 20). *Normalisasi Database: Definisi, Proses (Tahapan Lengkap) & Tujuan.* Retrieved from serupa.id: <https://serupa.id/normalisasi-database/>
- Yonata, J. (2023, Mei 2). *Pengertian MySQL, Fungsi, Cara Kerja, dan Kelebihannya.* Retrieved from dewaweb: <https://www.dewaweb.com/blog/apa-itu-mysql/>