**TUGAS SQL**

**KELOMPOK 10 :**

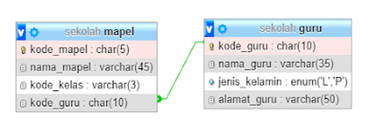
1. **MAHLIANA (200250502031)**
2. **KRISDAYANTO (200250502028)**

**Bab 4**

**Penggunaan perintah joins**

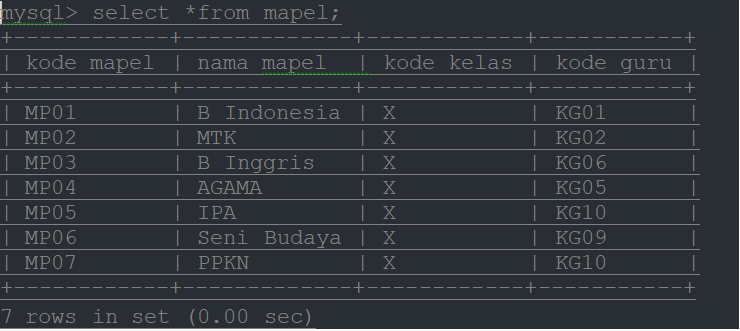
1. **Model Perintah join**

**Join** adalah penggabungan dua tabel yang saling terhubung atau terelasi antara colom dalam tabel satu dengan colom dalam tabel lain sehingga membentuk suatu informasi data yang lengkap atau menyeluruh. Sehingga kita dapat menampilkan data dari dua tabel yang berbeda menjadi satu tampilan / informasi dengan cara mengambil data colom yang sama walaupun colom yang lainnya berbeda perhatikan gambar berikut ini

****

Pada gambar diatas terdapat dua tabel yaitu tabel mapel dan juga tabel guru yang sudah terelasi atau terhubung melalui colom kode\_guru sebagai Primary Key ( dalam tabel Guru) dan colom kode\_guru sebagi Foreign Key ( dalam tabel mapel ).

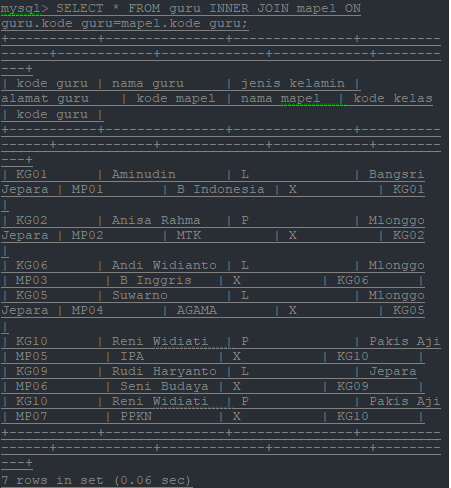
Kita tambahkan record atau data dari masing-masing tabel tersebut.

****

Dari kedua tabel diatas terdapat beberapa record data data dimana data yang ada di kolom kode guru dalam tabel yang fungsinya nanti digunakan untuk menghubungkan atau meraalisasikan data antara yang ada dalam tabel guru dan juga tabel mapel

1. **Penggabungan model inner join**

Menggabungkan data yang ada dalam tabel dengan inner join merupakan tipe penggabungan yang akan mengambil semua data dari tabel asal dan tabel tujuan dengan kondisi key yangterkait jika tidak maka data tersebut tidak akan muncul.Contoh:



Coba kalian lihat dari tabel diatas ada beberapa data guru yang tidak tampil dalam data diatas hanya menampilkan data guru yang memmpunyai realisasi dalam tabel mapel,data kode guru yang tidak terkait dengan tabel maka tidak ditampilkan.

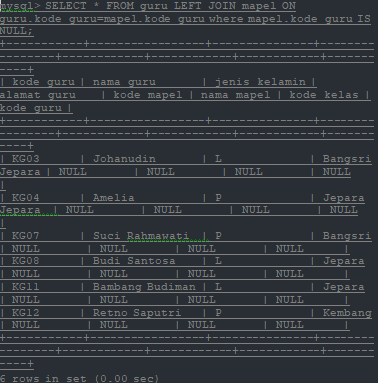
1. **Penggabungan Model Left join**

Perintah penggabungan left join bisa disebut juga dengan left outer join perintah ini digunakan untuk menampikan data sebalah kanan yang sesuai dengan kondisi penggabungan atau data yang cocok. Maka akan NULL Otomatis.



Dari data diatas menampilakan semua data dari kiri tabel guru secara lengkap dan didalam tabel mapel akan tersisi set NULL jika tidak ditemukan relasi data kode guru.

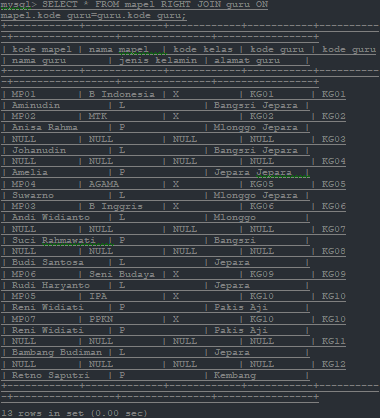
Kalian juga bisa menampilkan data guru yang belum mempunyai relasi/ NULL dengan tabel. Contoh querynya seperti dibawa ini.



1. **Penggabungan Model Right join**

Right join ini merupakan kebalikan dari left join, bisa disebut dengan right outer join yang mempunyai fungsi dia akan menampilkan data ditabel sebelah kanan dan mencari kecocokan relasi key pada tabel sebalah kiri. Jika tidak ditemukan kecocokan data relasi key maka secara otomatis akan null pada tabel sebelah kiri.

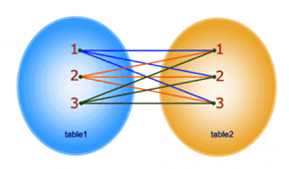
Contoh:



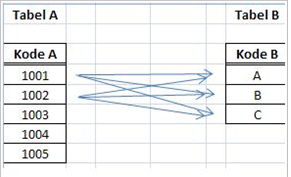
Pada hasil diatas terdapat null disebelah kiri karena tidak ditemukan kebocoran relasi key diantara tabel dan mepel guru.

1. **Pembangunan Model Cross join**

Cross Join, berfungsi mirip-mirip seperti inner join. Untuk lebih mudah memahaminya silahkan liat ilustrasi berikut:



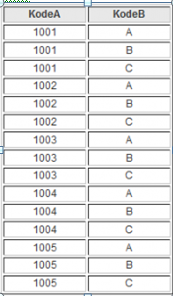
Misal saya memiliki table sebagai berikut:



SELECT kodeA ,kodeB FROM tabelA

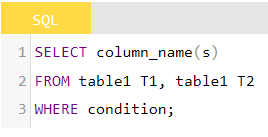
CROSS JOIN table ORDER BY KodeA

Maka akan memiliki hasil seperti ini:



1. **Penggabungan Model Selft join**

Self JOIN adalah gabungan reguler, tetapi tabel bergabung dengan dirinya sendiri.



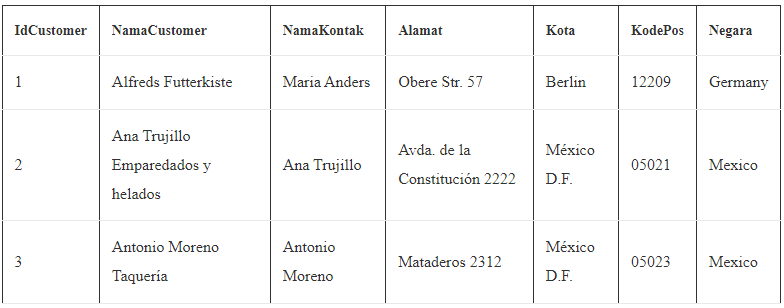
TI dan T2 adalah alias tabel yang berbeda untuk tabel yang sama.

* **Database Demo**

Dalam tutorial ini kita akan menggunakan database contoh Northwind

yang terkenal.

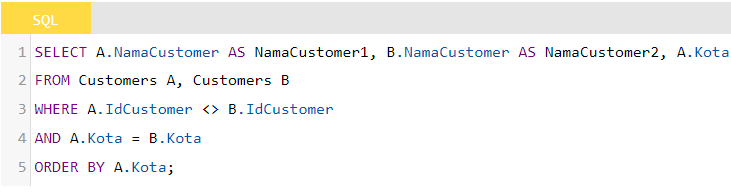
Di bawah ini adalah pilihan dari tabel “Customers”:



### Contoh SQL Self JOIN

Pernyataan SQL berikut cocok dengan pelanggan yang berasal dari

kota yang sama:



**Bab 15**

**Operasi String**

1. **LENGTH**

Operasi ini digunakan untuk menghitung panjang dari suatu *string.* Bentuku mumnya adalah **length** (*string*). Contoh:

Teks = “Demi kelestarianbumi, daur ulang sampah plastik”

Z = **length**(Teks)

Maka, Z = 48

Catatan:

(1) spasi dihitung karena dianggap karakter.

   (2) hasilnya juga berupa nilai numerik.

 (3) beberapa bahasa pemrograman menggunakan *statement*LEN( ).

Teks = “Demi kelestarianbumi, daurulangsampahplastik”

Z = **len**(Teks) – 10

      Maka, Z = 38

1. **CHAR\_LENGTH**

Fungsi CHAR\_LENGTH() dapat digunakan untuk mengembalikan panjang string (dalamkarakter).

Catatan: Fungsi ini sama dengan fungsi CHARACTER\_LENGTH().

**Syntax**

CHAR\_LENGTH(string)

Nilai Parameter

**Parameter Deskripsi**

string Required. String yang akan dihitung panjangnya

**Detail Teknis**

**Screenshot_34.png**

Contoh

**Kembalikanpanjang string:**

ELECT CHAR\_LENGTH("SQL Tutorial") AS LengthOfString;

Contoh

**Kembalikan panjang teks di kolom “NamaCustomer”:**

ELECT CHAR\_LENGTH (NamaCustomer) AS LengthOfName

FROM Customers;

1. **Hex**

berfungsi mengubah bilangan baik integer maupun float menjadi string hexadecimal. Bila bilangannya adalah float, kitaharus menggunakan fungsi float.hex() Sintaks

Fungsihex() memiliki sintaks sebagai berikut:

hex(x)

Parameter

Fungsihex() memiliki satu parameter yaitu:

x – bilangan yang akan diubah menjadi hexadecimal

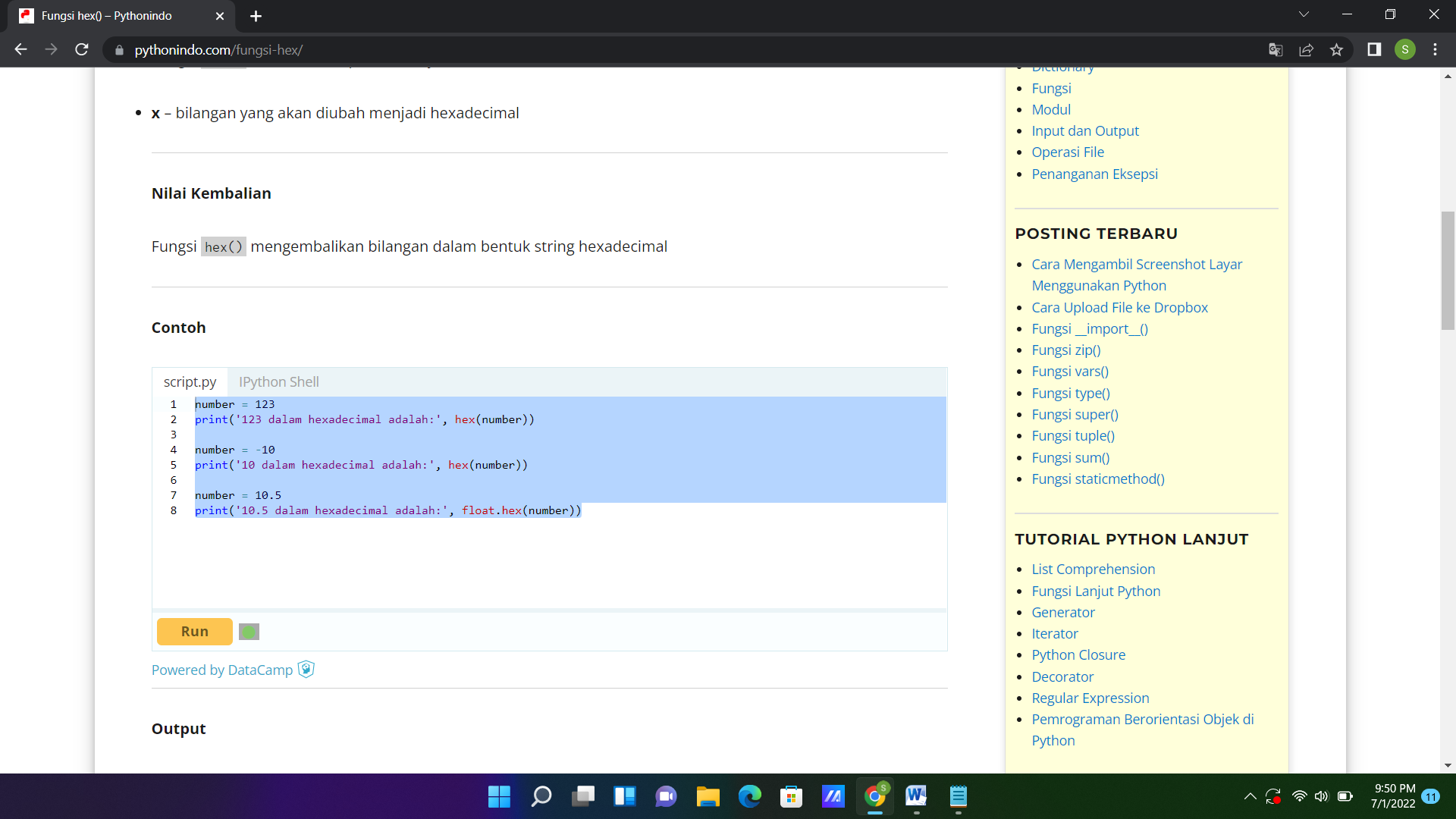
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NilaiKembalian

Fungsihex() mengembalikanbilangandalambentuk string hexadecimal

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Contoh



Output

Output dari program di atasadalahsepertiberikut:

123 dalam hexadecimal adalah: 0x7b

10 dalam hexadecimal adalah: -0xa

10.5 dalam hexadecimal adalah: 0x1.5000000000000p+3

1. **SUBSTRING**

Operasi ini paling sering dilakukan, sehingga fasilitas operasi ini disediakan di seluruh bahasa pemrograman.Bahkan di beberapa bahasa pemrograman, ada penggalan yang dilakukandarisebelahkiri, sebelahkanan, maupundarisembarangtempat.Namun, ada yang hanya menyediakan dari sembarang tempat saja, tidak khusus dari kiri atau kanan saja. Bentuk umum operasi ini adalah:

**substring(string, mulaikarakterkeberapa, banyaknyakarakterdiambil)**

Bentuk umum ini harus disesuaikan dengan kaidah daribahasa pemrograman yang kita gunakan.

Contoh:

      Teks = “Jangan ngaku beriman jika belum bisa menjaga

kebersihan lingkungan”

      Ambil1 = **substring**(Teks, 8, 13)

      Ambil2 = **substring**(Teks, 46, 10)

      Maka, nilaidari Ambil1 = “ngakuberiman” dan Ambil2 = “kebersihan”

Catatan: (1) spasidihitungkarenadianggapkarakter.

               (2) huruf besar dan kecil sesuai aslinya;

               (3) hasilnya juga berupa *string*;

(4) perhatikan bahasa pemrograman yang digunakan, ada yang memulai penghitungan

                   dari 0, bukan 1, danada yang menggunakan *statement* **copy**.

            Perhatikanjikadigabungdenganalgoritmasebagaiberikut:

Teks = “Safell Maya”

**FOR** x = 11 **TO** 1 **STEP** -1

**WRITE** **substring**(Teks, x, 1)

**NEXT** x

                Makahasilnya = “ayaMllefaS”

Teks = “C0B4-C0B4-D1B4C4”

**FOR** x = 1 **TO** 16 **STEP** 2

**WRITE** **substring**(Teks, x, 1)

**NEXT** x

                Makahasilnya = “CB-04DBC”

1. **UPPER/UCASE**

Mengembalikan Varian (String) yang berisi string yang ditentukan, yang dikonversikehurufbesar.

**Sintaks**

Ucase( string )

String yang diperlukan argumen adalah ekspresi string yang valid.Jika string berisi null, null dikembalikan.

Keterangan

Hanya huruf kecil yang dikonversi menjadi huruf besar; Semua huruf besar dan karakter non huruf tetap tidak berubah.

1. **STR TO DATE**

mengeksplorasi beberapa cara untuk mengubah objek String menjadi objek Date. Kita akan mulai dengan Date Time API baru, java.time, yang diperkenalkan di Java 8 sebelum melihat tipe data java.util.Date lama yang juga digunakan untuk merepresentasikan tanggal.

Untuk menyimpulkan, kita akan melihat beberapa perpustakaan eksternal untuk konversi menggunakan Joda-Time dan kelas Apache Commons Lang DateUtils.

Bacaan lebih lanjut:

Konversikan java.util.Date ke String

Pelajari beberapa metode untuk mengonversi objek Date menjadi objek String di Java.

Baca lebih lanjut →

Periksa Apakah String Adalah Tanggal yang Valid di Java

Lihat berbagai cara untuk memeriksa apakah sebuah String adalah tanggal yang valid di Java

Baca lebih lanjut →

Mengkonversi antara String dan Timestamp

Pelajari cara mengonversi antara String dan Timestamp dengan sedikit bantuan dari LocalDateTime dan Java 8.

Baca lebih lanjut →

* Mengonversi String ke LocalDate atau LocalDateTime

LocalDate dan LocalDateTime adalah objek tanggal-waktu yang tidak dapat diubah yang masing-masing mewakili tanggal, tanggal, dan waktu. Secara default, tanggal Java dalam format ISO-8601, jadi jika kita memiliki string yang mewakili tanggal dan waktu dalam format ini, maka kita dapat menggunakan parse() API dari kelas-kelas ini secara langsung.

Berikut sedikit lebih detail tentang API baru ini.

* Menggunakan Parse API

Date-Time API menyediakan metode parse() untuk mengurai String yang berisi informasi tanggal dan waktu. Untuk mengonversi objek String ke objek LocalDate dan LocalDateTime, String harus mewakili tanggal atau waktu yang valid menurut ISO\_LOCAL\_DATE atau ISO\_LOCAL\_DATE\_TIME.

Jika tidak, DateTimeParseException akan dilempar saat runtime.

Dalam contoh pertama kita, mari kita ubah sebuah String menjadi java.time.LocalDate:

Tanggal LocalDate = LocalDate.parse("05-05-2018");

Pendekatan serupa di atas dapat digunakan untuk mengonversi String ke java.time.LocalDateTime:

LocalDateTime dateTime = LocalDateTime.parse("05-05-2018T11:50:55");

Penting untuk dicatat bahwa objek LocalDate dan LocalDateTime adalah agnostik zona waktu. Namun, ketika kita perlu berurusan dengan tanggal dan waktu khusus zona waktu, kita dapat menggunakan metode parse ZonedDateTime secara langsung untuk mendapatkan waktu tanggal tertentu zona waktu:

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:dd z");

ZonedDateTime zonedDateTime = ZonedDateTime.parse("05-05-2015 10:15:30 Eropa/Paris", formatter);

Sekarang mari kita lihat bagaimana kita mengonversi string dengan format khusus.

* Menggunakan Parse API Dengan Formatter Kustom

Mengonversi String dengan format tanggal khusus menjadi objek Tanggal adalah operasi yang tersebar luas di Jawa.

Untuk tujuan ini, kita akan menggunakan kelas DateTimeFormatter, yang menyediakan banyak formatter yang telah ditentukan sebelumnya, dan memungkinkan kita untuk mendefinisikan formatter.

Mari kita mulai dengan contoh penggunaan salah satu formatter DateTimeFormatter yang telah ditentukan sebelumnya:

String dateInString = "19590709";

Tanggal LocalDate = LocalDate.parse(dateInString, DateTimeFormatter.BASIC\_ISO\_DATE);

Pada contoh berikutnya, mari kita buat formatter yang menerapkan format “EEE, MMM d yyyy.” Format ini menentukan tiga karakter untuk nama hari penuh dalam seminggu, satu digit untuk mewakili hari dalam sebulan, tiga karakter untuk mewakili bulan, dan empat digit untuk mewakili tahun.

Pemformat ini mengenali string seperti “Jumat, 3 Jan 2003″ atau “Rabu, 23 Mar 1994“:

String dateInString = "Senin, 05 Mei 1980";

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("EEE, d MMM yyyy", Locale.ENGLISH);

LocalDate dateTime = LocalDate.parse(dateInString, formatter);

* Pola Tanggal dan Waktu Umum

Mari kita lihat beberapa pola tanggal dan waktu yang umum:

• y – Tahun (1996; 96)

• M – Bulan dalam tahun (Juli; Juli; 07)

• d – Hari dalam sebulan (1-31)

• E – Nama hari dalam seminggu (Jumat, Minggu)

• a – penanda AM/PM (AM, PM)

• H – Jam dalam sehari (0-23)

• j – Jam dalam AM/PM (1-12)

• m – Menit dalam jam (0-60)

• s – Detik dalam menit (0-60)

Untuk daftar simbol lengkap yang dapat kita gunakan untuk menentukan pola untuk parsing klik di sini.

Jika kita perlu mengonversi tanggal java.time menjadi objek java.util.Date yang lebih lama, baca artikel ini untuk detail selengkapnya.

* Mengonversi String ke java.util.Date

Sebelum Java 8, mekanisme tanggal dan waktu Java disediakan oleh API lama dari kelas java.util.Date, java.util.Calendar, dan java.util.TimeZone, yang terkadang masih perlu kita kerjakan.

Mari kita lihat bagaimana mengubah sebuah String menjadi objek java.util.Date:

SimpleDateFormat formatter = new SimpleDateFormat("dd-MMM-yyyy", Locale.ENGLISH);

String dateInString = "7-Jun-2013";

Tanggal tanggal = formatter.parse(dateInString);

Dalam contoh di atas, pertama-tama kita perlu membuat objek SimpleDateFormat dengan melewatkan pola yang menjelaskan format tanggal dan waktu.

Selanjutnya kita perlu memanggil metode parse() dengan melewatkan String tanggal. Jika argumen String yang diteruskan tidak dalam format yang sama dengan pola, maka ParseException akan dilempar.

3.1. Menambahkan Informasi Zona Waktu ke java.util.Date

Penting untuk dicatat bahwa java.util.Date tidak memiliki konsep zona waktu, dan hanya mewakili jumlah detik yang berlalu sejak waktu epoch Unix – 1970-01-01T00:00:00Z.

Namun, ketika kita mencetak objek Date secara langsung, itu akan selalu dicetak dengan zona waktu sistem default Java.

Dalam contoh terakhir ini, kita akan melihat cara memformat tanggal sambil menambahkan informasi zona waktu:

SimpleDateFormat formatter = new SimpleDateFormat("dd-M-yyyy jj:mm:dd a", Locale.ENGLISH);

formatter.setTimeZone(TimeZone.getTimeZone("Amerika/New\_York"));

String dateInString = "22-01-2015 10:15:55";

Tanggal tanggal = formatter.parse(dateInString);

String formattedDateString = formatter.format(tanggal);

Kami juga dapat mengubah zona waktu JVM secara terprogram, tetapi ini tidak disarankan:

TimeZone.setDefault(TimeZone.getTimeZone("GMT"));

* Perpustakaan Eksternal

Sekarang setelah kita memiliki pemahaman yang baik tentang cara mengonversi objek String ke objek Date menggunakan API baru dan lama yang ditawarkan oleh inti Java, mari kita lihat beberapa library eksternal.

* Perpustakaan Joda-Time

Sebuah alternatif untuk inti Java Date and Time library adalah Joda-Time. Meskipun penulis sekarang merekomendasikan agar pengguna bermigrasi ke java.time (JSR-310), jika ini tidak memungkinkan maka perpustakaan Joda-Time memberikan alternatif yang sangat baik untuk bekerja dengan Tanggal dan Waktu. Pustaka ini menyediakan hampir semua kemampuan yang didukung dalam proyek Java 8 Date Time.

Artefak dapat ditemukan di Maven Central:

<ketergantungan>

<groupId>joda-time</groupId>

<artifactId>joda-time</artifactId>

<versi>2.10</versi>

</ketergantungan>

Berikut adalah contoh cepat bekerja dengan DateTime standar:

DateTimeFormatter formatter = DateTimeFormat.forPattern("dd/MM/yyyy HH:mm:dd");

String dateInString = "07/06/2013 10:11:59";

DateTime dateTime = DateTime.parse(dateInString, formatter);

Mari kita lihat juga contoh pengaturan zona waktu secara eksplisit:

DateTimeFormatter formatter = DateTimeFormat.forPattern("dd/MM/yyyy HH:mm:dd");

String dateInString = "07/06/2013 10:11:59";

DateTime dateTime = DateTime.parse(dateInString, formatter);

DateTime dateTimeWithZone = dateTime.withZone(DateTimeZone.forID("Asia/Kolkata"));

* Apache Commons Lang – DateUtils

Kelas DateUtils menyediakan banyak utilitas yang berguna, membuatnya lebih mudah untuk bekerja dengan objek Kalender dan Tanggal yang lama.

Artefak commons-lang3 tersedia dari Maven Central:

<ketergantungan>

<groupId>org.apache.commons</groupId>

<artifactId>commons-lang3</artifactId>

<version>3.12.0</version>

</ketergantungan>

Mari kita ubah String tanggal menjadi Java.util.Date menggunakan Array pola tanggal:

String dateInString = "07/06-2013";

Tanggal tanggal = DateUtils.parseDate(dateInString,

new String[] { "yyyy-MM-dd HH:mm:dd", "dd/MM-yyyy" });

1. **LOWER/LCASE**

Mengembalikan String yang telah dikonversi kehuruf kecil.

Sintaks

Lcase( string ) String yang diperlukanargumenadalahekspresistringyang valid.Jika ahstring mengandung Null, Null dikembalikan.

Keterangan

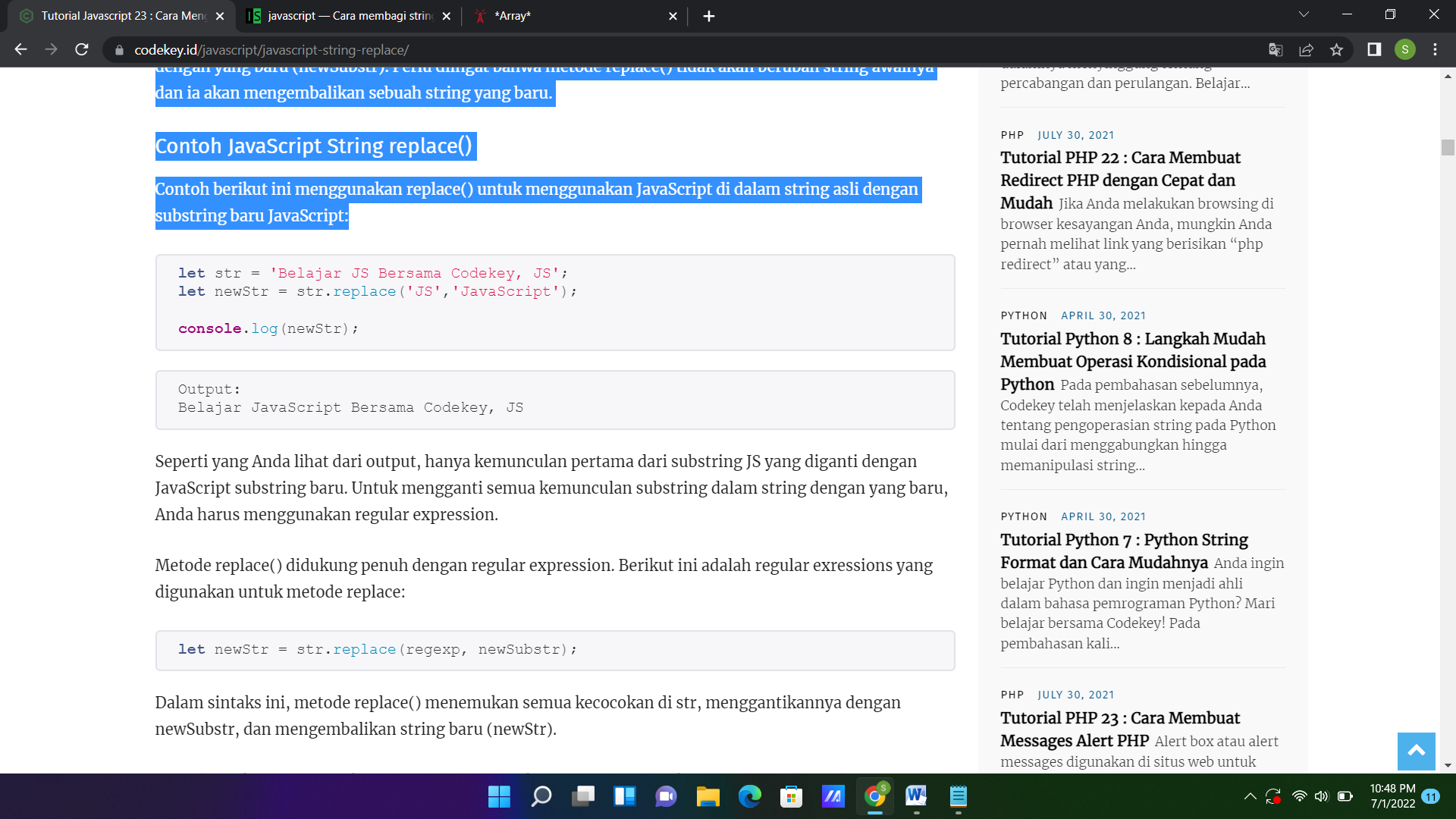
Hanya huruf besar yang dikonversi menjadi huruf kecil; Semua huruf kecil dan karakter nonhuruf tetap tidak beruba.

1. **REPLACE**

JavaScript replace() method mengembalikan string barudengansebuah substring (substr) yang diganikandengan yang baru (newSubstr). Perlu diingat bahwa metode replace() tidak akan berubah string awalnya dan ia akan mengembalikan sebuah string yang baru.

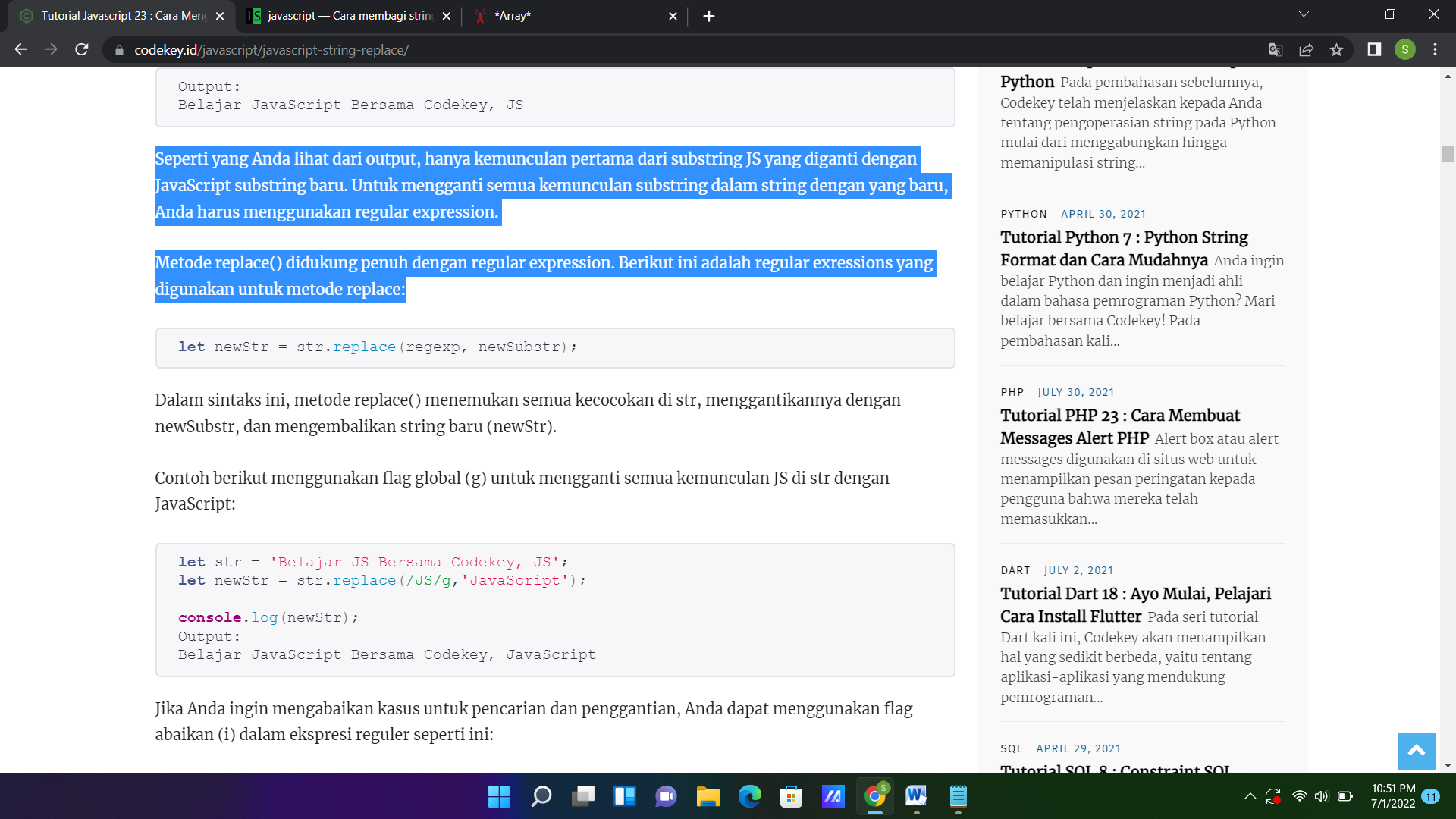
Contoh JavaScript String replace()

Contoh berikut ini menggunakan replace() untuk menggunakan JavaScript di dalam string asli dengan substring baru JavaScript:



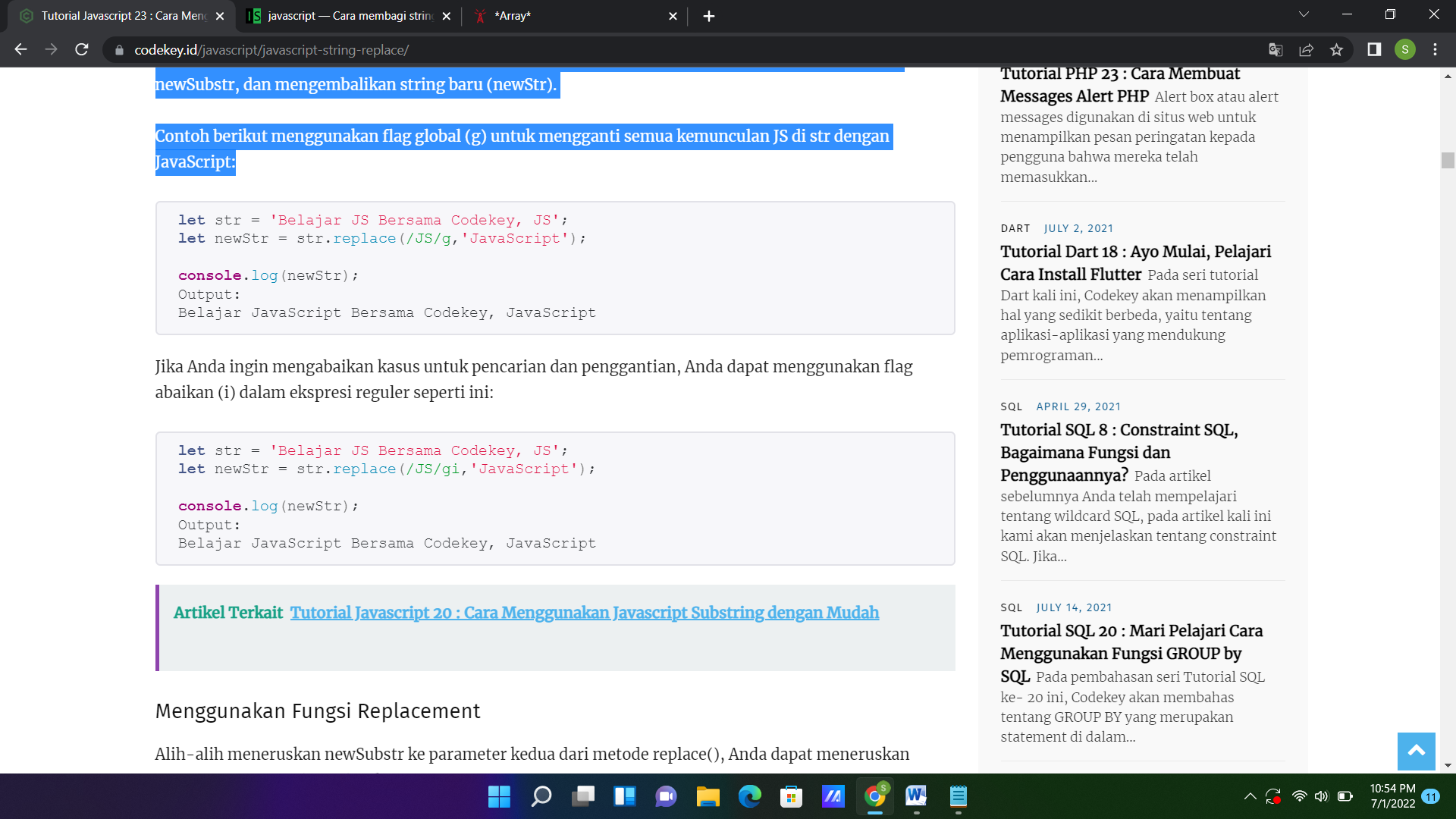
Seperti yang Anda lihat dari output, hanya kemunculan pertama dari substring JS yang diganti dengan JavaScript substring baru. Untuk mengganti semua kemunculan substring dalam string dengan yang baru, Anda harus menggunakan regular expression.

Metodereplace() didukung penuh dengan regular expression. Berikut ini adalah regular ssions yang digunakan untuk metode replace:

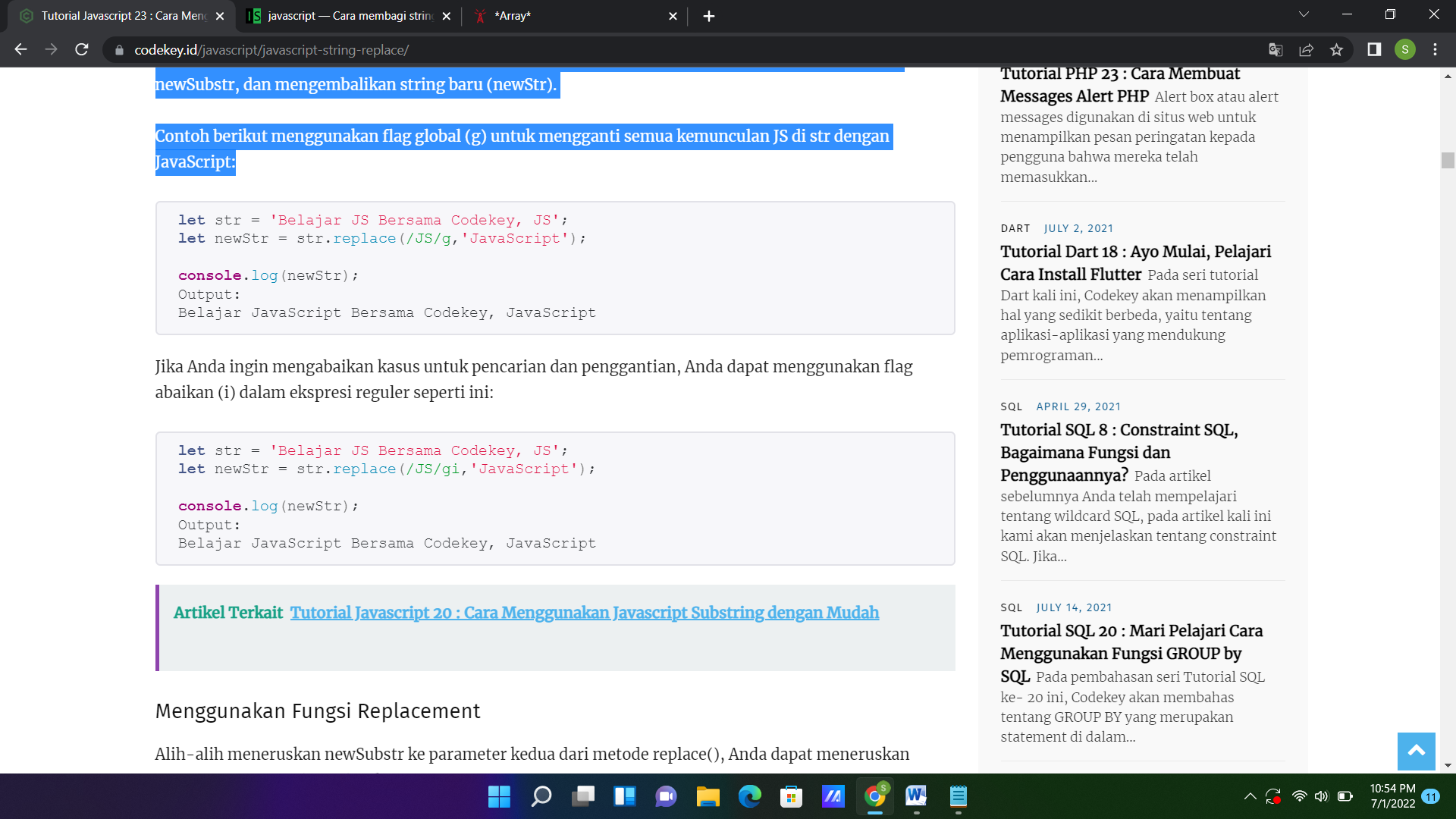


Dalam sintaks ini, metodereplace() menemukansemuakecocokan di str, menggantikannya dengan newSubstr, dan mengembalikan string baru (newStr).

Contoh berikut menggunakan flag global (g) untuk mengganti semua kemunculan JS di strdengan JavaScript:



Jika Anda ingin mengabaikan kasus untuk pencarian dan penggantian, Anda dapat menggunakan flag abaikan (i) dalam ekspresi reguler seperti ini:



* 1. **Mencari Elemen Dalam Daftar Yang Dipisahkan Koma**

Untuk mecari dan mengambil bebrapa kecocokan dalam daftar yang dipisahkan koma (dalam satu sel), anda dapat menggunakan fungsi if dengan fungsi text join.dalam contoh yang ditunjukkan, rumus di F5 adalah:

Screenshot_28.png

Ini adalah rumus array dan harus dimasukkan dengan control + shift + enter.

Rumus ini menggunakan rentang bernama "nama" (B5: B11) dan "grup" (C5: C11).

Inti dari rumus ini adalah fungsi IF, yang "memfilter" nama dalam tabel dengan warna seperti ini:

Screenshot_29.png

Tes logika memeriksa setiap sel dalam rentang bernama "grup" untuk nilai warna dalam E5 (merah dalam kasus ini). Hasilnya adalah array seperti ini:

Screenshot_30.png

Hasil itu digunakan bergantian untuk memfilter nama dari rentang bernama "nama":

Screenshot_31.png

Untuk setiap TRUE, nama bertahan, untuk setiap FALSE, IF mengembalikan string kosong ("").

Hasil IF terlihat adalah larik ini:

Screenshot_32.png

yang masuk ke fungsi TEXTJOIN sebagai teks1.

TEXTJOIN dikonfigurasi untuk menggunakan koma sebagai pembatas, dan untuk mengabaikan nilai kosong. Hasil akhirnya adalah string teks ini:

"Jude, Aya, George"

### Berbagai kondisi

Anda tidak dapat menggunakan fungsi DAN atau ATAU dalam rumus array seperti ini karena fungsi tersebut hanya mengembalikan satu hasil. Anda dapat menggunakan logika boolean seperti ini untuk AND:

Screenshot_33.png