

Study Kasus 1: Implementasi Algoritma Luhn

IN232 Matematika Diskrit

Latar Belakang

Umumnya, nomor kartu kredit memiliki 13, 15, atau 16 digit. Sebagai contoh,

$$4690358213754657 \quad (1)$$

merupakan nomor kartu kredit rekaan. Digit pertama menunjukkan sistem. Dalam nomor kartu kredit di Persamaan (1), digit pertama, 4, menunjukkan bahwa kartu tersebut adalah kartu Visa. Digit berikut menunjukkan informasi lain seperti nomor rekening dan nomor bank. (Penjelasan tepatnya tergantung pada jenis kartunya.) Digit terakhir adalah spesial karena digit tersebut dihitung dari digit sebelumnya dan disebut digit cek (*check digit*). Dalam Persamaan (1), digit cek adalah 7 dan nilai ini dihitung dari digit sebelumnya 4690 3582 1375 465. Digit cek kartu kredit digunakan untuk mengidentifikasi apakah nomor kartu kredit tersebut benar atau salah. Alasan penggunaan digit cek bukan untuk alasan keamanan, melainkan digit cek itu digunakan untuk membantu mendeteksi kesalahan seperti ketika seseorang memberikan nomor kartu kredit melalui telepon dan digit cek digunakan untuk mendeteksi apakah nomor kartu kredit yang diberikan tersebut benar atau salah (Johnsonbaugh, 2017).

Digit cek dihitung sebagai berikut. Mulai dari digit paling kanan; digit ini adalah digit cek dan kita skip digit ini. Gandakan setiap nomor lainnya. Jika hasil penggandaan adalah merupakan bilangan berdigit dua, tambahkan kedua digit tersebut menjadi satu angka; jika bukan bilangan berdigit dua, gunakan digit apa adanya.

4	6	9	0	3	5	8	2	1	3	7	5	4	6	5	
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	Double every other digit.
8	6	18	0	6	5	16	2	2	3	14	5	8	6	10	
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	Add digits of two-digit numbers.
8	6	9	0	6	5	7	2	2	3	5	5	8	6	1	

Figure 1: Contoh perhitungan algoritma Luhn

Jumlahkan angka yang dihasilkan

$$8 + 6 + 9 + 0 + 6 + 5 + 7 + 2 + 2 + 3 + 5 + 5 + 8 + 6 + 1 = 73.$$

Jika digit terakhir dari penjumlahan adalah 0, digit cek adalah 0. Jika digit terakhir bukan 0, kurangi 10 dengan digit terakhir untuk mendapatkan digit cek, contoh kasus di atas adalah $10 - 3 = 7$. Verifikasi digit cek pada kartu Visa, MasterCard, American Express, atau Diners Club favorit anda. Cara menghitung cek digit ini disebut *Algoritma Luhn*. Algoritma ini diciptakan oleh Hans Peter Luhn (1896-1964), yang menciptakannya saat di beliau bekerja di IBM. Meskipun awalnya dipatenkan, sekarang algoritma ini digunakan secara luas di domain publik.

Implementasi

Tugas anda adalah mengimplementasi Algoritma Luhn di dalam method

```

def compute_check_digit(a_number):
    """
    Compute a check digit from a sequence of credit card number by employing Luhn algorithm

    Parameters:
    -----
    a_number : int
        A credit card number. For example, 4690358213754657

    Returns:
    -----
    check_digit : int
        A digit between 0 dan 9 inclusively.
    """
    pass

```

Sebagai test drive, anda dapat coba memanggil

```

compute_check_digit(469035821375465) # 7
compute_check_digit(469035827375465) # 4

```

References

Johnsonbaugh, R. (2017). *Discrete Mathematics 8th Edition*. Pearson.