

# Pengujian *Correctness* Data Kartu Pembayaran pada Aplikasi *E-commerce* Menggunakan FitNesse

Fachri Veryawan Mahkota<sup>1</sup>, Eddy Prasetyo Nugroho<sup>2</sup>

*#Departemen Pendidikan Ilmu Komputer, Universitas Pendidikan Indonesia  
Bandung, Indonesia*

<sup>1</sup>mahkota@upi.edu

<sup>2</sup>eddyprn@upi.edu

**Abstract**—Along with technological advances, the use of e-commerce as a place to shop is increasingly attached to the daily lives of the wider community. E-commerce companies also take advantage of this phenomenon by providing applications for their e-commerce with various features, one of which is digital payments via credit or debit cards. This raises the risk of entering payment card data errors or fraudulent attempts with fake card data. Therefore, in this experiment the author tested the mock-up of the payment card validity testing function in an e-commerce application which after testing concluded it was good enough to detect correctness in payment card data, providing a safe guarantee for both e-companies, commerce and merchants who sell on the e-commerce platform.

**Keywords**—*payment card, testing, scenario-based, e-commerce, correctness*

**Abstrak**—Seiring dengan kemajuan teknologi, penggunaan e-commerce sebagai tempat berbelanja sudah semakin melekat dengan keseharian masyarakat luas. Para perusahaan e-commerce pun memanfaatkan fenomena ini dengan menyediakan aplikasi untuk e-commerce mereka dengan berbagai fitur yang salah satunya merupakan pembayaran digital melalui kartu kredit atau debit. Hal ini mendatangkan resiko adanya kesalahan pemasukan data kartu pembayaran ataupun upaya penipuan dengan data kartu palsu. Maka dari itu, pada percobaan ini Penulis melakukan *testing* kepada *mock-up* fungsi pengujian keabsahan kartu pembayaran di dalam aplikasi e-commerce yang setelah pengujian disimpulkan sudah cukup baik untuk mendeteksi *correctness* pada data kartu pembayaran, memberikan jaminan transaksi yang aman baik untuk perusahaan e-commerce maupun pelapak yang berjualan pada platform e-commerce tersebut.

**Kata kunci**—*kartu pembayaran, testing, scenario-based, e-commerce, keabsahan*

## I. PENDAHULUAN

Kemajuan teknologi – dalam hal ini internet – memungkinkan banyak orang melakukan hal-hal yang sebelumnya dirasa tidak mungkin untuk diwujudkan, salah satunya adalah berbelanja tanpa perlu menginjakkan kaki ke luar rumah. Pada awalnya, berbelanja secara daring hanya dapat dilakukan melalui komputer dengan cara

mengakses laman milik perusahaan e-commerce. Namun, semenjak ponsel pintar menjadi semakin terjangkau, banyak perusahaan e-commerce yang memperluas jangkauan mereka kepada calon pembeli potensial melalui aplikasi *mobile*.

Menurut Shankar V. dkk.[1], perangkat bergerak (*mobile*) yang selalu melekat dengan konsumen merupakan pintu gerbang hubungan antara konsumen dan penjual yang menjadikannya ideal untuk digunakan sebagai jalur untuk berjualan secara jarak jauh. Secara pembayaran pun konsumen dan penjual dimudahkan dengan adanya *cashless payment* (pembayaran non-tunai). Pembayaran non-tunai dapat dilakukan untuk membayar barang/jasa menggunakan kartu pembayaran tanpa memerlukan mata uang fisik[2].

Namun, perkembangan pada sektor e-commerce ini tidak serta merta hanya membawa dampak positif. Transaksi yang dilakukan serba daring ini memberikan celah kepada orang-orang yang tidak bertanggungjawab untuk melakukan penipuan, terutama penipuan pada aspek kartu pembayaran. Penipuan kartu pembayaran telah menyebabkan kerugian sebesar miliaran dolar dan juga telah mempengaruhi tingkat kepercayaan konsumen akibat penipuan tersebut[3]. Konsumen dalam hal e-commerce ini dapat mencakup pembeli maupun penjual.

Maka dari itu, perusahaan e-commerce perlu menerapkan sebuah mekanisme untuk memastikan bahwa kredensial kartu pembayaran yang dimasukkan pada saat proses *check-out* sudah betul atau absah. Mekanisme atau metode inilah yang Penulis angkat sebagai topik utama artikel ini dengan cara melakukan percobaan.

Pada percobaan ini, Penulis melakukan pengujian *correctness* data kartu pembayaran (debit/kredit Visa atau Mastercard) pada aplikasi e-commerce. Pengujian dilakukan pada aplikasi *mock-up* yang Penulis kembangkan menggunakan bahasa pemrograman Java sebagai representasi fungsi pengujian keabsahan data sebuah kartu pembayaran yang terdapat pada aplikasi e-commerce. Keabsahan kartu akan didasari oleh aturan penomoran kartu pembayaran yang telah ditetapkan oleh ISO.

Bagi perusahaan e-commerce, jika terjadi *error* pada fungsi pengujian keabsahan data kartu pembayaran, maka

akan muncul potensi penipuan menggunakan kartu pembayaran palsu yang akan menyebabkan kerugian materiil baik pada perusahaan *e-commerce* maupun pelapak yang berjualan pada platform *e-commerce* tersebut.

Pengujian akan dilakukan dengan FitNesse, dan langkah-langkah testing akan digambarkan menggunakan *scenario-based testing*.

**II. METODOLOGI**

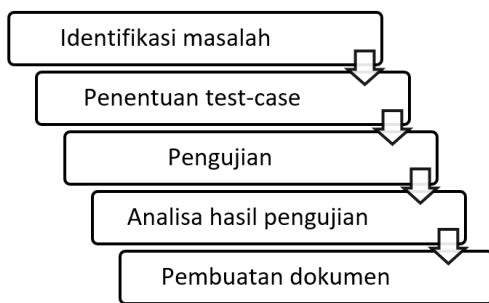
Cara mudah agar format makalah Anda sesuai dengan format makalah yang kami perlukan, gunakan dokumen ini sebagai *template* dan ketik teks Anda di dalamnya.

**A. Mendefinisikan Faktor Correctness Model McCall**

*Correctness* merupakan salah satu faktor dari model SQA McCall yang menganalisis akurasi hingga kelengkapan hasil output sebuah program[4].

Pada kasus ini, faktor *correctness* diujikan terhadap data kartu pembayaran yang didapatkan dengan elemen nomor kartu pembayaran, kode CVV (*Card Verification Value/Code*) dst. Proses ini biasanya terjadi pada sisi *back-end* dari sebuah aplikasi *e-commerce*.

**B. Mendefinisikan Skenario Pengujian**



Gambar. 1 Diagram *scenario-based testing*

Skenario pengujian yang dirancang sebagaimana tercantum di atas menggambarkan langkah-langkah *scenario-based testing*[5][6] sebagai berikut:

- 1) *Identifikasi Masalah*: Menentukan masalah yang akan diangkat sebagai fokus pengujian. Pada kasus ini, masalah yang diangkat adalah *correctness* data kartu pembayaran pada aplikasi *e-commerce*.
- 2) *Penentuan Test-case*: Menetapkan contoh masukan dan keluaran program yang diharapkan pada saat pengujian. Pada kasus ini, *test-case* merupakan nomor kartu pembayaran (*card number*) dan nama pemilik kartu pembayaran (*cardholder name*).
- 3) *Pengujian*: Melakukan pengujian dengan alat pengujian yang telah ditetapkan. Pada kasus ini, pengujian dilakukan dengan menggunakan FitNesse.
- 4) *Analisa Hasil Pengujian*: Melakukan analisa terhadap hasil pengujian yang dikeluarkan oleh alat pengujian. Analisa dari *output* pengujian yang dilakukan

oleh FitNesse tertera pada bab ketiga (hasil dan pembahasan).

- 5) *Pembuatan Dokumen*: Membuat dokumen mengenai persiapan, proses, dan hasil pengujian yang telah dilakukan. Seluruh persiapan, proses, dan hasil pengujian telah dipaparkan pada artikel ini.

**C. Menetapkan Syarat Keabsahan Data Kartu Pembayaran**

Keamanan sebuah kartu pembayaran bergantung kepada keamanan fisik dari kartu tersebut (*chip*, PIN, NFC, dll.) juga *privacy* dari kartu tersebut[7]. Aspek-aspek keamanan ini menjadi acuan awal keabsahan data sebuah kartu pembayaran, yang kemudian dapat ditelusuri lebih lanjut dengan memperhatikan informasi lain yang tertera pada kartu tersebut mulai dari nomor kartu hingga CVV (*Card Verification Value/Code*). CVV dapat membandingkan identitas kartu dengan data yang dimiliki oleh bank penerbit kartu.

Menurut ISO/IEC 7812-1:2017[8] mengenai sistem penomoran kartu pembayaran (dalam hal ini kartu kredit/debit), maka ditetapkan bahwa prasyarat keabsahan kartu dalam proses testing meliputi sebagai berikut:

- Mengandung 16 digit nomor kartu yang terpisah oleh spasi setiap empat digit (*card number*).
- Memiliki nama pemilik kartu minimal dua suku kata (*cardholder name*).

**D. Menentukan Alat dan Spesifikasi Pengujian**

Alat pengujian yang akan dipakai pada percobaan ini merupakan FitNesse. FitNesse[9] merupakan web server, *wiki*, dan *automated testing tool* untuk perangkat lunak yang berbasis pada *Framework for Integrated Testing* milik Ward Cunningham dan dirancang untuk mendukung *acceptance testing*.

Pengujian menggunakan FitNesse dilakukan pada dua buah *class* Java, yaitu:

- 1) *Class Checkname*: Digunakan untuk menguji nama pemilik kartu. Menerima masukan berupa *string* dan keluaran berupa *string* “Nama valid” atau “Nama tidak valid”.
- 2) *Class Checkcc*: Digunakan untuk menguji nomor kartu pembayaran. Menerima masukan berupa *string* dan keluaran berupa *string* “Nomor kartu valid” atau “Nomor kartu tidak valid”.

Pengujian dilakukan dengan menentukan *test case* dan *expected value* terlebih dahulu. Untuk mendapatkan *expected value* berupa “Nama/nomor kartu valid” akan dibutuhkan *input* nama/nomor kartu yang memenuhi syarat keabsahan data kartu pembayaran sebagaimana telah dicantumkan pada poin sebelumnya. Maka dari itu, *script* pengujian FitNesse akan dibangun menggunakan aturan sebagai berikut:

- 1) *Class Checkname: Value* yang diberikan harus sama atau lebih dari 2. Pada kasus ini, diharapkan nama

yang diberikan memiliki minimal dua suku kata sebagai syarat dasar nama depan dan nama belakang.

3) *Class Checkcc*: *Value* yang diberikan harus sama dengan 4. Pada kasus ini, diharapkan pada nomor kartu yang diberikan terdapat empat bagian nomor kartu sebagai syarat keabsahan kartu tersebut.

III. HASIL DAN PEMBAHASAN

```
public class Checkname {
    private String name;
    public void setName(String name) {
        this.name = name;
    }

    public String result() {
        ArrayList<String> spl = new
        ArrayList<String>(Arrays.asList(name.split("
        ")));
        ...
    }
}
```

Gambar. 2 Kode class Checkname

```
public class Checkcc {
    private String cc;
    public void setCc(String cc) {
        this.cc = cc;
    }

    public String result() {
        ArrayList<String> spl = new
        ArrayList<String>(Arrays.asList(cc.split("
        ")));
        ...
    }
}
```

Gambar. 3 Kode class Checkcc

Kedua kode class Java di atas merupakan hasil pengembangan dari spesifikasi aplikasi (class) kandidat pengujian yang tertera di poin metodologi. Pada masing-masing class, dilakukan pemecahan input yang diberikan menggunakan fungsi split dengan patokan karakter spasi. Hasil split kemudian dihitung jumlahnya. Jika jumlahnya memenuhi akan didapatkan output “valid”, dan sebaliknya akan mendapatkan output “tidak valid”.

TABEL I  
TEST CASE CLASS CHECKNAME

No.	Input	Output
1	Fachri Veryawan Mahkota	Nama valid
2	Adithya Kurniawan	Nama valid
3	Argi	Nama tidak valid
4	Albari Berki Pradhana	Nama valid
5	Alfauzan	Nama tidak valid
6	Misael	Nama tidak valid
7	Dhimas Satria Hanandyatama	Nama valid
8	Muhammad Ramadhan Maulana A	Nama valid
9	Syachrul	Nama tidak valid
10	Jonathan	Nama tidak valid

TABEL III  
TEST CASE CLASS CHECKCC

No.	Input	Output
1	4827 3546 1828 4612	Nomor kartu valid
2	5999 3728 1010 4763	Nomor kartu valid
3	4888 47371 828 9090	Nomor kartu tidak valid
4	4999 0321 5629 3746	Nomor kartu valid
5	50293847 7894 2636	Nomor kartu tidak valid
6	5878 4636 4777	Nomor kartu tidak valid
7	5655 7474 3526 4019	Nomor kartu valid
8	5777 3829 5616 2535	Nomor kartu valid
9	4099 57382737 1636	Nomor kartu tidak valid
10	4999 5889 46371059	Nomor kartu tidak valid

Untuk membangun script pengujian FitNesse, dibutuhkan set test-case dari masing-masing class. Pada kedua tabel di atas terdapat 10 buah test-case—input dan output (expected value)—untuk masing-masing class.

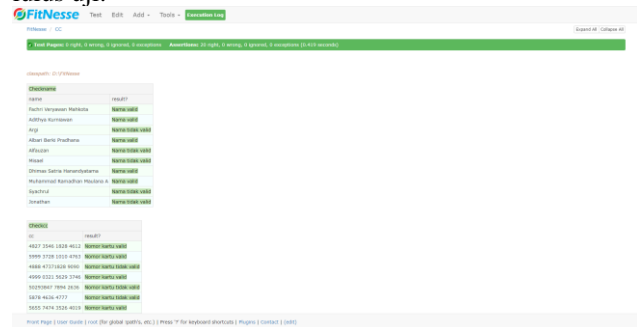
```
!*< Hidden
!define TEST_SYSTEM {slim}
*!

!path D:\FitNesse

|Checkname|
|name|result?|
|Fachri Veryawan Mahkota|Nama valid|
|Adithya Kurniawan|Nama valid|
```

Gambar. 4 Script FitNesse yang dibangun dari test-case yang diberikan

Script FitNesse yang telah dibangun akan memberikan instruksi kepada FitNesse untuk melakukan pengujian dengan patokan: class yang akan diuji, variabel yang harus di set, dan output (result) yang didapatkan. Jika variabel yang di set dan output yang diberikan oleh class sama, maka dapat dikatakan bahwa bagian dari pengujian tersebut lulus uji.



Gambar. 5 Antarmuka FitNesse

FitNesse menggunakan antarmuka web yang di host secara lokal pada komputer. Setelah memasukkan script dan memulai pengujian, tampilan antarmuka FitNesse terlihat seperti gambar di atas.

Checkname	
name	result?
Fachri Veryawan Mahkota	Nama valid
Adithya Kurniawan	Nama valid
Argi	Nama tidak valid
Albari Berki Pradhana	Nama valid
Alfauzan	Nama tidak valid
Misael	Nama tidak valid
Dhimas Satria Hanandyatama	Nama valid
Muhammad Ramadhan Maulana A	Nama valid
Syachrul	Nama tidak valid
Jonathan	Nama tidak valid

Gambar. 6 Hasil pengujian *test-case* pada *class* Checkname

Checkcc	
cc	result?
4827 3546 1828 4612	Nomor kartu valid
5999 3728 1010 4763	Nomor kartu valid
4888 47371828 9090	Nomor kartu tidak valid
4999 0321 5629 3746	Nomor kartu valid
50293847 7894 2636	Nomor kartu tidak valid
5878 4636 4777	Nomor kartu tidak valid
5655 7474 3526 4019	Nomor kartu valid
5777 3829 5616 2535	Nomor kartu valid
4099 57382737 1636	Nomor kartu tidak valid
4999 5889 46371059	Nomor kartu tidak valid

Gambar. 7 Hasil pengujian *test-case* pada *class* Checkcc

Berdasarkan hasil pengujian pada kedua *class* menggunakan FitNesse, dapat disimpulkan bahwa program memenuhi *expected value* yang telah ditetapkan dan tidak terdapat *error* pada saat program dijalankan. Pada saat *input* data yang diberikan valid, maka *output* yang diberikan menunjukkan bahwa data yang diberikan valid, begitupun sebaliknya.

#### IV. KESIMPULAN

Percobaan ini menilai kualitas aplikasi *e-commerce* dalam memastikan keabsahan data dari kartu pembayaran—baik kredit maupun debit—yang digunakan pada proses pembayaran pada saat bertransaksi. Keabsahan berdasar pada standar identitas kartu pembayaran yang telah ditetapkan oleh ISO.

Keluaran yang diharapkan dari pengujian masing-masing *class* (Checkname untuk mengecek nama pemilik kartu dan Checkcc untuk mengecek nomor kartu) adalah nama pemilik kartu dan nomor kartu yang valid dengan aturan nama pemilik lebih dari dua suku kata dan nomor kartu memiliki jumlah 16 digit yang terpisah dengan spasi setiap empat digit.

Perlu diperhatikan juga bahwa pengujian yang dilakukan pada percobaan ini masih dijalankan pada program *mock-up* dikarenakan aplikasi yang tidak bersifat *open-source*. Pada dasarnya, pengujian akan jauh lebih efektif jika dilakukan pada aplikasi yang sesungguhnya dan dilakukan juga setelah aplikasi tersebut melewati tahap pengembangannya.

#### V. SARAN

Percobaan ini diharapkan dapat menjadi *stepping stone* dalam pengujian selanjutnya yang dilakukan pada aplikasi *e-commerce* yang riil baik oleh pihak internal perusahaan *e-commerce* maupun oleh pihak eksternal. Pengujian akan memberikan hasil yang lebih relevan dengan kasus sehari-hari jika dilakukan pada aplikasi yang sesungguhnya, baik dalam tahapan pengembangan dan juga pasca pengembangan.

#### UCAPAN TERIMA KASIH

Penulis menyampaikan segala puji dan syukur kepada Tuhan Yang Maha Esa atas kelancaran dalam percobaan dan penulisan artikel ini. Penulis juga berterima kasih kepada orang tua dan keluarga Penulis yang memberikan dukungan moril serta teman-teman seperjuangan yang telah menjadi sarana bertukar pendapat Penulis hingga terselesaikannya artikel ini.

#### REFERENSI

- [1] Shankar, V., Venkatesh, A., Hofacker, C., & Naik, P. (2010). Mobile marketing in the retailing environment: current insights and future research avenues. *Journal of interactive marketing*, 24(2), 111-120.
- [2] Ryman-Tubb, N. F., Krause, P., & Garn, W. (2018). How Artificial Intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark. *Engineering Applications of Artificial Intelligence*, 76, 130-157.
- [3] Sakharova, I. (2012, June). Payment card fraud: Challenges and solutions. In 2012 IEEE international conference on intelligence and security informatics (pp. 227-234). IEEE.
- [4] Galin, D. (2004). *Software quality assurance: from theory to implementation*. Pearson Education India.
- [5] Tsai, W. T., Na, Y., Paul, R., Lu, F., & Saimi, A. (n.d.). Adaptive scenario-based object-oriented test frameworks for testing embedded systems. *Proceedings 26th Annual International Computer Software and Applications*.
- [6] Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave macmillan.
- [7] Raj, S. B. E., & Portia, A. A. (2011, March). Analysis on credit card fraud detection methods. In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET) (pp. 152-156). IEEE.
- [8] ISO/IEC 7812-1:2017, <https://www.iso.org/standard/70484.html>.
- [9] FitNesse Web Site, <http://www.FitNesse.org>.