

PERANCANGAN *BACK-END* APLIKASI E-KOST DENGAN MODEL WATERFALL BERBASIS WEB

BACK-END DESIGN APPLICATION SOFTWARE OF E-KOST WITH WATERFALL MODEL

Nofrialdi Dwi Putra¹, Ir. Burhanuddin Dirgantoro M.T², Ratna Astuti Nugrahaeni S.T.,M.T³

^{1,2,3}Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom
¹nofrialdidp@students.telkomuniversity.ac.id, ²burhanuddin@telkomuniversity.ac.id,
³ratnaan@telkomuniversity.ac.id

Abstrak

Informasi kost yang ada saat ini belum memberikan informasi yang memuaskan calon penyewa. Banyak calon penyewa yang harus mendatangi sendiri ke lokasi karena ketidak lengkapan dari informasi yang diberikan. Untuk penyewa yang berasal dari luar kota akan sangat kesulitan untuk mencari tempat kost. Kesulitan tersebut disebabkan karena banyaknya kost-kostan sekitar yang belum di ketahui oleh penyewa. Daerah yang akan dijadikan tempat penelitian adalah *Telkom University*. Metode perancangan program aplikasi yang digunakan adalah *waterfall* dengan pendekatan berorientasi objek menggunakan pemodelan *UML*. Apapun teknik pengumpulan data yang digunakan yaitu observasi, wawancara dan studi pustaka bahasa pemrograman javascript sebagai *web programming* dan *MySQL* sebagai *database*. Aplikasi E-Kost ini akan menampilkan tempat-tempat kost yang disewakan beserta harga, fasilitas, promosi dan informasi peta lokasi rumah kost yang dikehendaki. Penyewa dapat mencari informasi berdasarkan lokasi sehingga calon penyewa yang berasal dari luar kota mendapatkan informasi tempat kost sesuai dengan kebutuhan. Pemilik kost dapat mengiklankan tempat kostnya dengan mendaftar sebagai pemilik dalam sistem ini. System ini dilakukan testing menggunakan *unit test* dan *blackbox*, yang nantinya mempengaruhi seberapa *valid* data dan meminimalisir *bug-bug* yang ada. E-Kost juga memiliki fitur untuk iklan-iklan yang bisa diisi oleh semua orang yang mau mempublikasikan productnya.

Kata kunci System Informasi , Website , Javascript, Mysql

Abstract

Boarding information that is currently available does not provide information that satisfies prospective tenants. Many prospective tenants must come to the location on their own due to the incompleteness of the information provided. For important tenants from out of town it will be very difficult to find a boarding house. This difficulty is due to the many boarding houses that have not been found by the tenants. The area that will be used as a research site is Telkom University. The design method of application program used is a waterfall with the objective of object oriented using UML modeling. Whatever data collection techniques used are observation, interviews and the study of the javascript programming language as web programming and MySQL as a database. This E-Boarding application will display boarding places that are rented through prices, facilities, promotions and map information of the desired boarding house location. Tenants can find information about the location so that prospective tenants who are needed from outside the city get boarding information as needed. Boarding owners can advertise their boarding places by registering as an owner in this system.

Keywords : *Information System, Website, Javascript, Mysql*

1. Pendahuluan

Pada saat ini teknologi sudah menjadi salah satu kebutuhan pokok bagi banyak orang untuk mendapatkan semua informasi. Semakin berkembangnya zaman banyak orang yang memilih jalan cepat untuk mendapatkan informasi salah satu contoh adalah informasi untuk mencari kost bagi para perantau[1].

Masalah yang terjadi saat ini untuk informasi terkait kost masih sangat sulit dikarenakan informasi yang diberikan masih kurang lengkap, serta menghindari adanya calo selain itu juga untuk pemesanan kost saat ini harus datang ke lokasi sehingga membutuhkan waktu yang lama untuk proses mendapatkan kost.

Permasalahan setiap orang jika ingin mencari tempat menginap atau kost adalah tidak mau dipusingkan dengan berbagai hal terutama ribet dan kurangnya informasi tentang kost yang ingin dicari dan sesuai dengan kriteria yang diinginkan[2].

Maka dari itu dengan adanya aplikasi dengan judul “PERANCANGAN BACK-END APLIKASI E-KOST DENGAN MODEL WATERFALL BERBASIS WEB”, ini akan memudahkan para pencari kost untuk mendapatkan kost yang diinginkan. Banyak fitur seperti foto kost, harga, lokasi kamar dan kelengkapan kamar serta rekomendasi unit. Aplikasi ini juga memudahkan pemilik kost untuk mengontrol atau melihat dan juga mempublikasikan kost yang dimilikinya di aplikasi ini[3].

2. Dasar Teori

2.1 Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. Perkembangan sistem informasi telah menyebabkan terjadinya perubahan yang cukup signifikan dalam pola pengambilan keputusan yang dilakukan oleh manajemen baik pada tingkat operasional.

Perkembangan ini juga telah menyebabkan perubahan-perubahan peran dari para manajer dalam pengambilan keputusan, mereka dituntut untuk selalu dapat memperoleh informasi yang paling akurat dan terkini. Meningkatnya penggunaan teknologi informasi, khususnya internet, telah membawa setiap orang dapat melaksanakan berbagai aktivitas dengan lebih akurat, berkualitas, dan tepat waktu[6].

Setiap organisasi dapat memanfaatkan internet dan jaringan teknologi informasi untuk menjalankan berbagai aktivitasnya secara elektronik. Pernyataan di atas dapat disimpulkan bahwa suatu sistem informasi merupakan suatu perangkat kerja yang dapat bekerja untuk memproses suatu masukan ataupun data, kemudian data yang telah diproses tersebut akan diproses dan menjadi suatu keluaran yang berguna untuk mencapai suatu tujuan tertentu[3].

2.1 Express Js

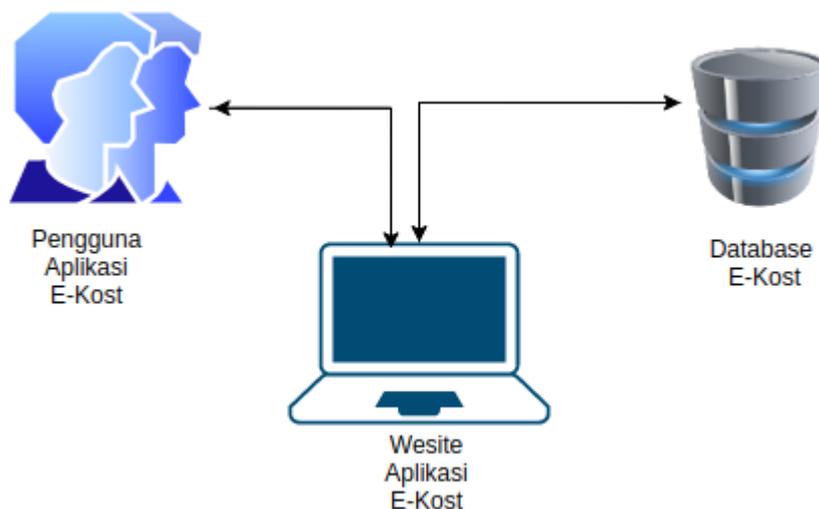
Express js adalah sebuah framework javascript yang digunakan dari sisi back-end untuk membangun sebuah aplikasi web. Framework ini sangat banyak digunakan di kalangan developer untuk membangun web dari sisi back-end. Framework ini menawarkan beberapa fitur seperti routing, rendering view dan mendukung middleware dengan kata lain anda akan banyak menghemat waktu dalam pengembangan aplikasi Node.js[11].

Expressjs merupakan framework minimal yang sangat fleksibel. Anda bisa membuat web server HTML, server file statik, aplikasi chat, search engine, sosial media, layanan web dengan akses melalui REST API atau aplikasi hybrid yaitu selain pengguna mempunyai akses melalui REST API juga mempunyai akses ke HTML page. Untuk membangun sebuah website dengan cepat dan efisien, express js adalah salah satu jawabannya. Karena banyak dilengkapi dengan library bawaan dan dokumentasi yang sangat jelas dari website penyediaanya.

3. Pembahasan

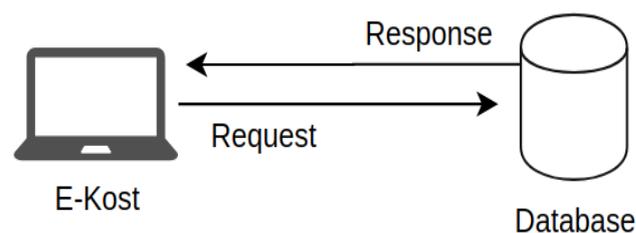
Beberapa fungsi-fungsi di backend yang diimplementasikan pada aplikasi E-kost ini seperti search kosan, add kosan, edit kosan serta pemesanan kosan tersebut dan ditambah dengan *connecting* dengan *databasenya*. Karena *back-end* adalah segala proses yang terjadi di belakang web maka dari itu untuk implementasi dibawah akan menampilkan sebagian source code yang mana itu adalah implementasi dari beberapa fitur-fitur di aplikasi E-kost.

3.1 Perancangan Sistem



Gambar 1- Gambaran Umum Sistem E-Kost

Gambar diatas adalah gambaran umum sistem kerja website aplikasi E-Kost. User atau pengguna melakukan view pada tampilan yang sudah dibangun oleh front-end. Dan ketika user melakukan request atau view salah satu kosan artinya user melakukan request terhadap aplikasi, barulah back-end melakukan response balik ke website aplikasi tersebut.



Gambar 2- Sistem E-Kost (Backend)

Karakteristik pengguna pada aplikasi E-Kost ini yaitu ada admin, pencari kost dan pemilik kost. Masing-masing pengguna memiliki hak yang berbeda-beda karena masing-masing juga fungsionalitanya berbeda.

1. Admin

Hak dari admin adalah semua yang berkaitan dari aplikasi tersebut, mulai dari admin bisa melakukan apa yang dilakukan oleh pemilik kost, serta admin juga bisa melihat apa yang dilakukan oleh si pengguna atau si pencari kost. Namun bedanya adalah admin bisa melakukan approve kost juga si pencari sudah melakukan pemesanan kost serta sudah melakukan pembayaran juga. Disinilah admin melakukan approve pemesanan dan langsung mengirimkan notifikasi ke pemilik kost dan si pemesan atau pencari kost.

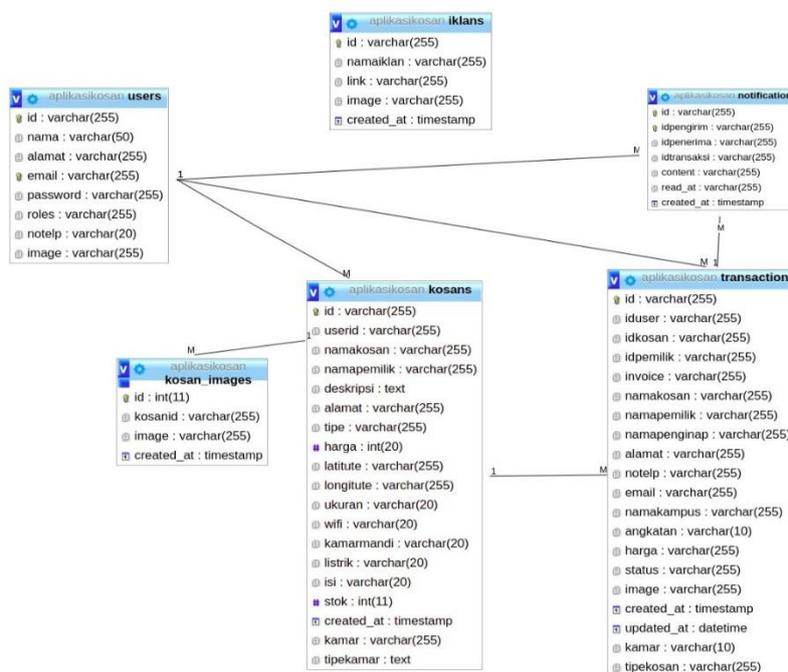
2. Pemilik Kost

Hak dari pemilik kost adalah melihat siapa aja yang menginap di kostannya, serta pemilik kost juga bisa menambahkan atau mengurangi list kost yang ada di aplikasi E-Kost. Pemilik kost juga ada fungsionalitas untuk melihat notifikasi yang dikirim dari admin jika ada pemesanan kost.

3. Pencari Kost

Pencari kost memiliki hak seperti melakukan view kost pada halaman aplikasi E-Kost. Pencari juga bisa melakukan pemesanan kost serta memilih kamar mana yang ingin dicari, serta melakukan pembayaran dengan pihak ketiga atau bisa dibidang pihak ketiga adalah admin. Nanti ketika sudah dibayar admin akan melakukan approve dan notifikasi ke pencari kost akan muncul beserta nomor invoice bukti pembayaran yang nantinya akan dibawa ke kost tujuannya.

3.2 Diagram Entity



Gambar 3- Diagram Entity

4. Pengujian dan Analisis

4.1 Implementasi

4.1.1 Implementasi Fungsionalitas

Beberapa fungsi-fungsi di backend yang diimplementasikan pada aplikasi E-kost ini seperti search kosan, add kosan, edit kosan serta pemesanan kosan tersebut dan ditambah dengan *connecting* dengan *databasenya*. Karena *back-end* adalah segala proses yang terjadi di belakang web maka dari itu untuk implementasi dibawah akan menampilkan sebagian source code yang mana itu adalah implementasi dari beberapa fitur-fitur di aplikasi E-kost.

- Koneksi Database
- Hapus Kos
- Tambah Kos
- Pencarian
- Ubah Kos
- Iklan

4.1.2 Implementasi Metode Waterfall

Table 1- Metode Waterfall

No	Tahapan	Uraian
1	Analisis <i>Requirement</i>	Analisis kebutuhan dilakukan dengan cara melakukan sosialisasi dengan mahasiswa baru atau lama yang nantinya akan mencari kost baru serta para pemilik kost serta orang-orang sekitar lokasi ruang lingkup aplikasi ini
2	Desain	Perancangan desain menggunakan beberapa diagram seperti ERD, UML dan lain lain yang juga berkolaborasi dengan <i>front-end</i> hal-hal apa saja yang nantinya perlu ada di aplikasi ini.
3	Implementasi	Sistem aplikasi E-Kost ini dibuat dengan menggunakan Bahasa pemrograman javascript dan platform Nodejs untuk backend. Serta database menggunakan Mysql dan session database menggunakan Redis-server
4	Pengujian Sistem	Pengujian ini dilakukan pada aspek-aspek tertentu, terlebih lagi backend tidak langsung berhubungan dengan user. Maka dari itu pengujian dilakukan dengan <i>Unit Testing</i> . Yang nantinya mengecek fungsionalitas code yang ada pada aplikasi apakah berjalan dengan baik dan meminimalisir <i>bug-bug</i> yang ada
5	Maintenance	Maintenance dilakukan ketika aplikasi sudah siap dijalankan. Untuk saat ini maintenance juga dilakukan, ketika dilakukan testing ada bug maka disitulah maintenance aplikasi dilakukan

4.2 Pengujian

4.2.1 Pengujian *Black Box*

Table 2- Pengujian inputan login register

Kasus dan Hasil Uji			
Data Masukan	Harapan	Hasil Pengamatan	Kesimpulan
Inputan login berupa username(email) dan password	Data yang dikirim fornt-end dilakukan pengecekan di db dan session aktif	Data di cek dan session aktif ketika true	Valid
Inputan register	Data yang dikirim ketika melakukan register tersimpan ke database	Data tersimpan ke database mysql	Valid

Table 3-Pengujian inputan Kosan

Kasus dan Hasil Uji			
Data Masukan	Harapan	Hasil Pengamatan	Kesimpulan
Inputan kosan baru	Data inputan akan disave ke database	Data tersimpan	Valid
Tambah Gambar kos	Image name tersimpan ke directori yang sudah ditentukan dan <i>filename</i> tersimpan ke database	File tersimpan kedirectory dan filename tersimpan	Valid
Ubah data kos	Data perubahan disimpan ke database	Data tersimpan ke database	Valid
Hapus kos	Data di database akan dihapus	Data kos berhasil dihapus	Valid

Table 4-Pengujian Pemesanan Kos

Kasus dan Hasil Uji			
Data Masukan	Harapan	Hasil Pengamatan	Kesimpulan
Inputan Transaksi Pemesanan	Data inputan pemesanan akan tersimpan ke table transaksi	Data berhasil disimpan	Valid
Inputan bukti pembayaran	Data bukti pembayaran akan di simpan ke database	Data disimpan	Valid

Table 5-Pengujian Tambah Iklan

Kasus dan Hasil Uji			
Data Masukan	Harapan	Hasil Pengamatan	Kesimpulan
Inputan data-data iklan	Data iklan tersimpan dan bisa ditampilkan	Data iklan tersimpan serta bisa ditampilkan	Valid

4.2.2 Pengujian Unit

```

GET /dashboard/profile/09212 302 4.594 ms - 33
PASS ./unit.test.js
Unit Test Aplikasi E-Kost
Get index (/) Halaman Utama
  ✓ respond berhasil untuk load tampilan index / (608 ms)
Get Login (/auth/login)
  ✓ respond Berhasil untuk load tampilan login (14 ms)
register (/auth/register)
  ✓ respond berhasil untuk load tampilan register (10 ms)
login (/auth/login)
  ✓ respond berhasil untuk login dan data cek ke db (24 ms)
  ✓ respond berhasil untuk login gagal tanpa email (9 ms)
Register (/auth/register)
  ✓ respond berhasil untuk register berhasil (361 ms)
  ✓ respond berhasil untuk login gagal tanpa email (308 ms)
Dashboard Overview (/dashboard/overview)
  ✓ respond berhasil untuk masuk ke dashboard (10 ms)
Dashboard Add Kosan (/dashboard/addkosan)
  ✓ respond berhasil untuk masuk ke dashboard (9 ms)
Add Kosan (/dashboard/addkosan)
  ✓ respond berhasil Kosan berhasil ditambahkan (63 ms)
Dashboard Kosan (/dashboard/kosan)
  ✓ respond berhasil untuk masuk ke dashboard (7 ms)
Dashboard Edit Kosan (/dashboard/editkosan)
  ✓ respond berhasil untuk masuk ke dashboard (39 ms)
Dashboard Transaksi (/dashboard/transaksi)
  ✓ respond berhasil untuk masuk ke dashboard transaksi (15 ms)
Dashboard user (/dashboard/user)
  ✓ respond berhasil untuk masuk ke dashboard user (7 ms)
Dashboard Pemilik (/dashboard/pemilik)
  ✓ respond berhasil untuk masuk ke dashboard pemilik (9 ms)
Dashboard transaksi admn (/dashboard/transaksiadmin)
  ✓ respond berhasil untuk masuk transaksistadmin (55 ms)
Dashboard profile (/dashboard/profile)

```

Gambar 4- Hasil Pengujian Unit - 1

```

unit.js | 100 | 100 | 100 | 100 |
ibuntuSoftware/ | 100 | 100 | 100 | 100 |
connection.js | 100 | 100 | 100 | 100 |
aplikasikost/routes | 24.4 | 10.43 | 32.79 | 26.49 |
auth.js | 55.56 | 25 | 66.67 | 56.34 | 15-22,45-67,104-108,116-118
index.js | 16.25 | 7.37 | 18.22 | 15.99 | 26-60,115-310,322-329,327,331-428,445-506,526-570,582
notifications.js | 57.89 | 25 | 54.55 | 57.89 | 11-26,38,52,62-67
users.js | 80 | 100 | 0 | 80 | 6
aplikasikost/routes/dashboard | 24.38 | 7.66 | 12.6 | 25.47 |
addkosan.js | 40.40 | 33.04 | 37.5 | 39.24 | 9-19,23-30,38-42,63-65,74-76,85-140
editkosan.js | 15.79 | 3.7 | 9.09 | 17.14 | 13-20,29-57,75-135,139-179
iklan.js | 33.96 | 10 | 25 | 34.62 | 28-35,47-54,60-88
kosan.js | 17.65 | 7.14 | 8.33 | 18.75 | 8-18,21-50,55-63,76-110,135-147,157-160
overview.js | 40 | 16.67 | 25 | 43.9 | 18-57
pemilik.js | 30.19 | 14.29 | 14.29 | 31.37 | 9-35,45-53,72-84
profile.js | 30.77 | 8.33 | 11.11 | 31.37 | 11-14,23-30,47-64,74-89
transaksi.js | 12.69 | 2.38 | 3.33 | 13.44 | 110,131-144,163-184,194-205,210-232,245-259,274-281,293-296,308-363
transaksistadm.js | 19.11 | 3.33 | 9.09 | 20.41 | 13-23,26-39,52-75,83-94,110-112,117-132,145-160,177-184,219-274
transaksipemilik.js | 35.9 | 6.25 | 16.67 | 38.03 | 11-22,37-45,71-72,88-102,117-124
user.js | 48.28 | 25 | 25 | 50 | 24-30,52-60
aplikasikost/uttl | 37.18 | 0 | 0 | 37.18 |
nall.js | 27.78 | 0 | 0 | 27.78 | 9-48
upload.js | 40 | 0 | 0 | 40 | 7-10,18,24-30
uploadbukt.js | 40 | 0 | 0 | 40 | 7-10,18,24-30
uploadiklan.js | 40 | 0 | 0 | 40 | 7-10,18,24-30
uploadprofile.js | 40 | 0 | 0 | 40 | 7-10,18,24-30
aplikasikost/views/helpers | 40 | 35.71 | 25 | 43.9 |
index.js | 40 | 35.71 | 25 | 43.9 | 16-38,48-66,73-74
-----
Test Suites: 1 passed, 1 total
Tests: 49 passed, 49 total
Snapshots: 0 total
Time: 4.965 s
Ran all test suites.
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?

```

Gambar 5- Hasil Pengujian Unit - 2

5. Kesimpulan dan Saran

5.2 Kesimpulan

Berdasarkan hasil pengujian yang sudah dilakukan sebelumnya dapat disimpulkan bahwa:

1. Sistem Back-end sudah berjalan dengan semestinya sesuai dengan yang diharapkan
2. Pemakaian framework Express js sangat bisa dipahami apalagi untuk diintegrasikan dengan *front-end*, dikarenakan sama-sama menggunakan bahasa pemrograman yang sama yaitu javascript.

3. Semua fungsionalitas berjalan dengan baik dan dapat membantu pengguna untuk mengakses web secara cepat dan dinamis.
4. Berdasarkan unit testing, source code dari back-end cukup bagus dan bisa berjalan dengan baik.

5.2 Saran

Hasil dari tugas akhir ini mungkin sangat jauh dari kata sempurna dan masih banyak juga kekurangannya. Untuk itu untuk perlu pengembangan lagi dari sisi backend dan source code nya biar *best practice* dan *clean code*. Beberapa hal penting yang perlu dikembangkan lagi adalah:

1. Fitur integrasi dengan google maps
2. Login integrasi dengan Google Api login
3. serta fitur-fitur lainnya.

6. Daftar Pustaka

- [1] hakim, "sistem informasi pencarian kontrakan dan rumah kost kota surabaya," *skripsi universitas muhamadiyah surabaya*, p. 1, 2011.
- [2] I. P. T. K. Bakti Abidin1, "SISTEM INFORMASI RUMAH KOST ONLINE BERBASIS WEB DAN MESSAGING," p. 4.
- [3] I. P. K. Bakti Abidin, "Sistem Informasi Rumah Kost Online Berbasis Web dan Messaging," 2012.
- [4] I. K. Shinta Siti Sundari*1, "PERANCANGAN SISTEM INFORMASI RUMAH KOST BERBASIS WEB DAN SHORT MESSAGE SERVICE (SMS) MENGGUNAKAN PHP DAN MySQL," *Seminar Nasional Informatika 2015*, 2015.
- [5] 2. O. R. B. B. L. F. G. C. M. A. Ayres1, "Back-end Limitations in Advanced Nodes and Alternatives".
- [6] J. J. X. D. H. S. Kun Liua *, "Design and Development of Management Information System for Research Project Process Based on Front-end and Back-end Separation," *2017 International Conference on Computing Intelligence and Information System*, 2017.
- [7] Sastra,Putra Nababan, "Apakah Javascript akan menjadi masa depan aplikasi web dan mobile", 2016, <https://medium.com/@sastranababan/apakah-javascript-akan-menjadi-masa-depan-aplikasi-web-dan-mobile-e0015299809f>
- [8] Guntoro, "Metode Waterfall : Pengertian, Tahapan, Contoh, Kelebihan dan Kekurangan", 2020, <https://badoystudio.com/metode-waterfall/>
- [9] Chinedu, Orié, "Testing NodeJs/Express API with Jest and Supertest", 2019 , dev.to/nedsoft/testing-nodejs-express-api-with-jest-and-supertest-1km6
- [10] Kurniawati, Peni , "Pengujian Sistem", 2018 , <https://medium.com/skyshidigital/pengujian-sistem-52940ee98c77>
- [11] Setiadi, Fikri, "Membuat CRUD dengan Node js , Bootstrap dan Mysql", 2018, <http://mfikri.com/artikel/crud-nodejs-mysql>