

# Pembangunan *Backend* Pada *Website* Aplikasi Fateka: Forum Alumni Teknik Komputer Universitas Telkom

1<sup>st</sup> Muhammad Armain Hidayatullah  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
muhammadarmain@student.telkomuni-  
versity.ac.id

2<sup>nd</sup> Anggunmeka Luhur Prasasti  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
anggunmeka@telkomuniversity.ac.id

3<sup>rd</sup> Faisal Candrasyah Hasibuan  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
faicanhasfcb@telkomuniversity.ac.id

**Abstrak**—Teknologi mengalami perkembangan yang sangat pesat dalam beberapa tahun terakhir ini, hal ini menyebabkan meningkatnya sistem informasi dan komunikasi di berbagai bidang, mulai dari akses data, informasi aktual, iklan, komunikasi, dan bidang pendidikan. Perguruan tinggi pasti memerlukan kemajuan teknologi dalam kegiatan peningkatan kualitas pelayanan informasi dalam menyajikan informasi dan komunikasi secara luas. Aplikasi Fateka hadir menyediakan informasi dan media interaksi untuk menjemati civitas akademika dengan seluruh alumni Teknik Komputer Universitas Telkom. Fateka sebagai sebuah *website* aplikasi yang di dukung oleh fitur-fitur utamanya seperti data alumni, lowongan karir, media interaksi, kegiatan, berita, artikel dengan tampilan yang ramah pengguna. Dengan hadirnya aplikasi Fateka, diharapkan mempermudah akses pengguna maka dibutuhkan aplikasi *backend* yang akan menyediakan dan mengolah berbagai data yang akan terjadi pada *platform* Fateka yang berbasis *website*. Pada *website* aplikasi Fateka menggunakan dua metode pengujian yaitu *unit testing* dan *stress testing*, hasil dari *unit testing* menggunakan *framework* laravel dan *library* laravel *phpunit* menunjukkan bahwa fitur dapat berjalan dengan baik dan berdasarkan hasil pengujian *stress testing* menggunakan *apachebench* dengan mengirimkan 1000 *request* ke *server*, dengan hasil yang didapatkan waktu yang dibutuhkan untuk *testing* 292.423 *seconds*, *complete requests* 1000, *concurrency level* 100, dan total rata-rata waktu yang dibutuhkan dibawah 5 detik.

**Kata kunci**— alumni, *backend*, laravel, sistem informasi.

## I. PENDAHULUAN

Teknologi mengalami perkembangan yang sangat pesat dalam beberapa tahun terakhir ini, hal ini menyebabkan meningkatnya sistem informasi dan komunikasi di berbagai bidang, mulai dari akses data, informasi aktual, iklan, komunikasi, dan bidang pendidikan. Perguruan tinggi pasti memerlukan kemajuan teknologi dalam kegiatan peningkatan kualitas pelayanan informasi dalam menyajikan informasi dan komunikasi secara luas. Salah satunya adalah pemanfaatan jejaring sosial untuk kepentingan media pengumpulan data alumni [1].

Penelitian atau produk aplikasi yang mewadahi alumni sudah dibuat dilembaga lain, contohnya FAST untuk alumni Telkom dan Ikatan Alumni untuk ITB keduanya memiliki kesaamaan yang didalamnya menampilkan layanan informasi seputar universitas dan kegiatan alumninya. Kekurangan dari produk aplikasi diatas tidak adanya informasi lowongan

kerja, dan koneksi antara mahasiswa dengan alumni terbatas. Dibuatnya aplikasi Fateka untuk menjaga kontinuitas koneksi dengan pihak kampus, para alumni sering membagikan lowongan karir yang pendistribusiannya dilakukan secara parsial. Sehingga informasi tersebut tidak menyebar secara luas di lingkungan Teknik Komputer. Dikarenakan belum ada portal khusus dari Program Studi yang berfungsi untuk memfasilitasi berbagai keperluan seperti informasi lowongan kerja, dan tambahan lain seperti layanan berita dan artikel, kegiatan sharing alumni, dan ruang virtual yang dapat digunakan untuk kegiatan pertemuan secara online. Selain itu juga pendataan dan penelusuran alumni, guna memudahkan dalam hal pengolahan data alumni tersebut.

Aplikasi Fateka dibangun pada *platform website*, dengan hadirnya aplikasi Fateka diharapkan mempermudah akses pengguna maka dibutuhkan aplikasi *backend* yang akan menyediakan dan mengolah berbagai data yang akan terjadi pada *platform* Fateka yang berbasis *website*.

## II. KAJIAN TEORI

### A. Sistem Informasi

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Informasi adalah data yang telah di olah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun masa yang akan datang [2]. Sistem informasi adalah kumpulan komponen dan adanya informasi yang berkaitan untuk mencapai suatu tujuan tertentu. Sistem informasi terdiri dari beberapa komponen. Komponen-komponen sistem informasi terdiri dari *hardware*, *software*, data prosedur, dan manusia [3].

### B. Backend

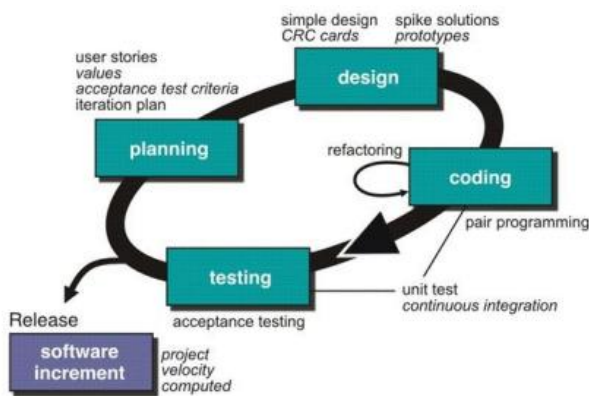
Dalam Pada pengembangan web, kebanyakan *framework* menggunakan model arsitektur MVC (*Model View Controller*). Model mengandung fungsi-fungsi yang berhubungan dengan database, seperti *create*, *read*, *update*, dan *delete*. *Backend* adalah tempat di mana proses pada suatu sistem informasi atau aplikasi berjalan, data dapat ditambahkan, diubah maupun dihapus. *Backend* biasanya mengurus segala jenis proses yang tidak berhubungan langsung dengan pengguna, seperti *server* dan basis data.

*Backend* dibutuhkan dalam pengembangan sistem dan manajemen data pada sistem [4].

### C. Extreme Programming

Merupakan salah satu konsep pengembangan dari pendekatan agile. Metode *Extreme Programming* (XP) memberikan percepatan perencanaan pada setiap perubahan kebutuhan fungsi-fungsi yang dibutuhkan *client*, sehingga memenuhi semua kebutuhan karena melalui tahapan yang iterative dan incremental [5]. *Extreme Programming* ini memiliki beberapa nilai yang dijadikan landasan yakni *Communication, Simplicity, Feedback* dan *Courageness*.

*Extreme Programming* (XP) dikembangkan oleh Kent Beck yang menurut penjelasannya "*Extreme Programming* adalah metode pengembangan *software* yang cepat, efisien, berisiko rendah, fleksibel, terprediksi, *scientific* dan menyenangkan". Metode XP adalah metode responsif terhadap perubahan dimana iterasi dapat dilakukan berulang kali sesuai dengan kebutuhan. XP menggunakan tahapan waktu yang singkat dan mengulang untuk bagian-bagian yang berbeda sesuai target yang dicapai [6]. Tahapan-tahapan XP dapat dilihat pada gambar 1 berikut ini.



GAMBAR 1  
Fase pada *Extreme Programming*

Berdasarkan gambar 1 maka fase dalam metodologi pengembangan sistem XP adalah sebagai berikut:

#### 1. *Planning* (Perencanaan)

Tahapan yang pertama dalam membuat sistem dilakukan perencanaan seperti mengidentifikasi masalah, menganalisa kebutuhan sampai pada tahap penetapan jadwal pembangunan sistem.

#### 2. *Design* (Perancangan)

Pada tahapan ini melakukan perancangan pemodelan yang dimulai dari pemodelan sistem, pemodelan arsitektur sampai pemodelan basis data. Pemodelan sistem dan arsitektur menggunakan diagram UML sedangkan database menggunakan ERD.

#### 3. *Coding* (Pengkodean)

Pada tahapan ini melakukan penerapan pemodelan yang sebelumnya sudah dibuat dalam bentuk user interface dengan menggunakan bahasa pemrograman.

#### 4. *Testing* (Pengujian)

Pada tahapan ini pengujian sistem dilakukan untuk mengetahui kesalahan apa yang terjadi pada aplikasi yang dikembangkan, serta mengetahui sistem yang dibangun sudah sesuai dengan kebutuhan yang telah ditentukan.

### D. Unit Testing

Unit Testing adalah pengujian terautomasi yang dilakukan dengan cara melakukan isolasi terhadap suatu program atau aplikasi menjadi unit kecil untuk menguji kesesuaian data masukan dan keluaran agar sesuai dengan hasil yang diharapkan. Pengujian unit ditulis dan dieksekusi oleh pengembang perangkat lunak untuk memastikan kode sesuai dengan persyaratan dan seperti yang diharapkan. Karena tujuan dari pengujian unit adalah memastikan setiap bagian dari program bekerja dengan benar, maka setiap fungsi atau kode ketika diuji harus mengembalikan nilai-nilai yang tepat [7].

### E. Stress Testing

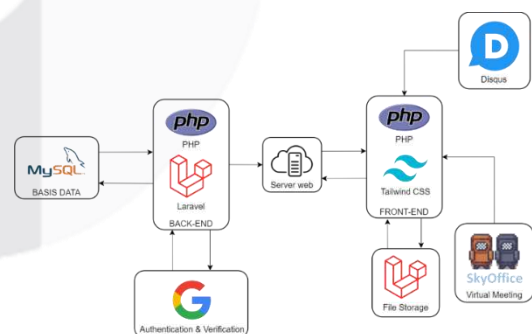
*Stress testing* merupakan pengujian perangkat lunak untuk menguji stabilitas dan kemampuan sistem dalam menerima *request* yang banyak. Pengujian ini menentukan seberapa kuat sistem jika sedang diakses oleh banyak *user*. Dalam *stress testing* nantinya akan dilakukan simulasi pengujian menggunakan total *request* dan *request* yang diuji dalam satu waktu yang sama. Tujuannya adalah untuk menganalisis perilaku sistem setelah terjadi kegagalan.

*Stress testing* adalah proses yang digunakan untuk mengevaluasi kinerja suatu sistem atau komponen dalam kondisi ekstrim. Ini dapat mencakup beban tinggi, suhu ekstrim, atau kondisi lain yang mensimulasikan skenario dunia nyata [8].

## III. METODE

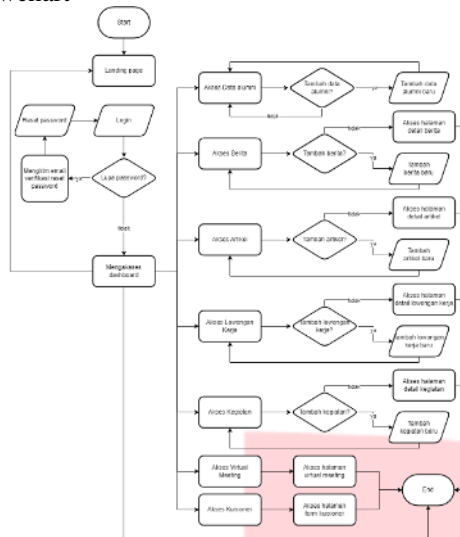
### A. Arsitektur Website Fateka

Komponen utama aplikasi Fateka terdiri dari *backend* dan *frontend*, selain itu juga terdapat beberapa integrasi dengan pihak ketiga seperti Google sebagai media penghubung untuk melakukan *authentication* dan *verification*, disqus yang dimanfaatkan sebagai media diskusi antara dosen, alumni, dan mahasiswa, laravel *file storage* sebagai tempat menyimpan media, skyoffice dimanfaatkan sebagai media *virtual meeting* antara dosen, alumni, dan mahasiswa.



GAMBAR 2  
Arsitektur lengkap *website* aplikasi Fateka

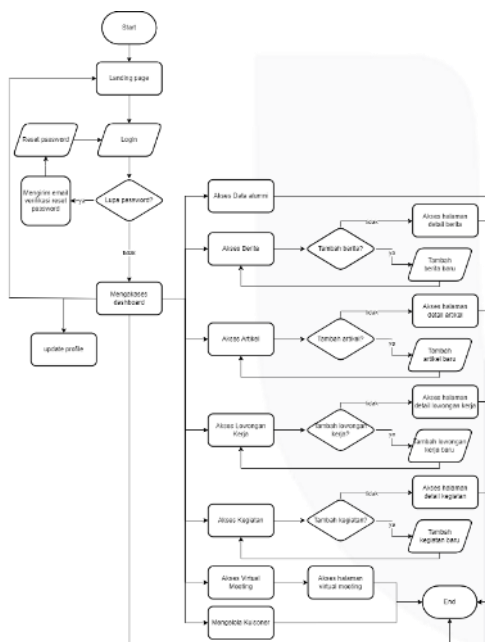
## B. Flowchart



GAMBAR 3

Flowchart user dengan role admin

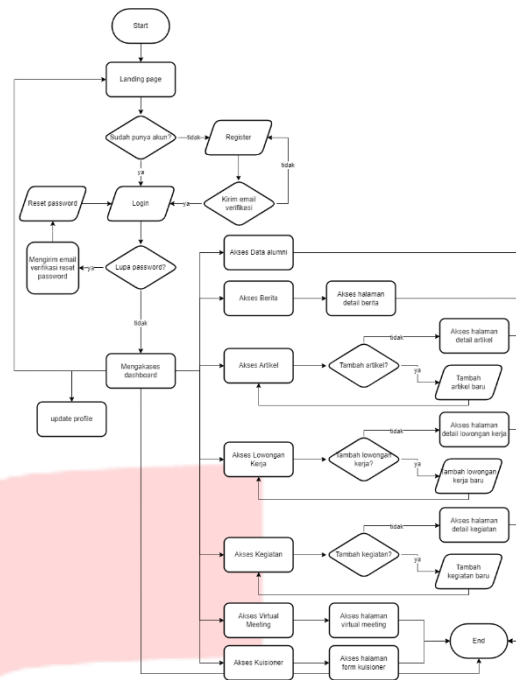
Gambar 3 adalah alur pengguna dengan role admin untuk masuk kedalam dashboard, pengguna akan masuk ke dashboard dan melakukan beberapa tahapan-tahapan sesuai dengan *roles* dan *permissions*.



GAMBAR 4

Flowchart user dengan role dosen

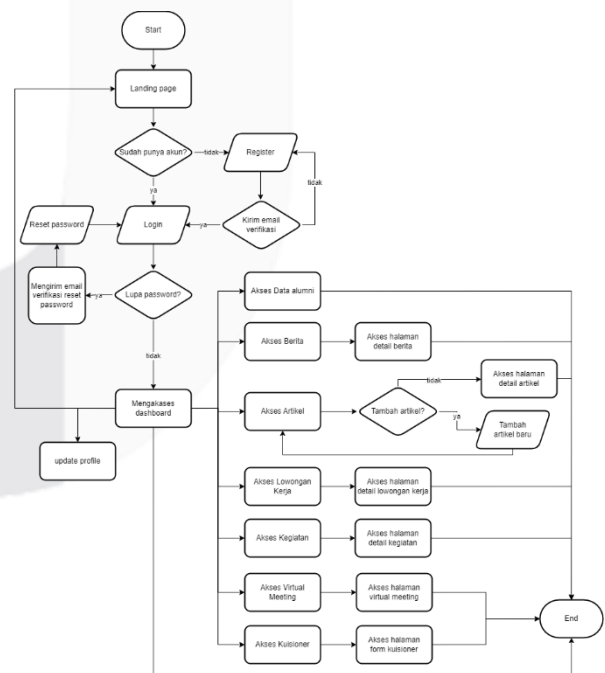
Gambar 4 adalah alur pengguna dengan role dosen untuk masuk kedalam dashboard, pengguna akan masuk ke dashboard dan melakukan beberapa tahapan-tahapan sesuai dengan *roles* dan *permissions*.



GAMBAR 5

Flowchart user dengan role alumni

Gambar 5 adalah alur pengguna dengan role alumni untuk masuk kedalam dashboard pengguna akan melakukan registrasi terlebih dahulu jika belum memilih akun lalu pengguna akan masuk ke dalam dashboard dan melakukan beberapa tahapan-tahapan sesuai dengan *roles* dan *permissions*.



GAMBAR 6

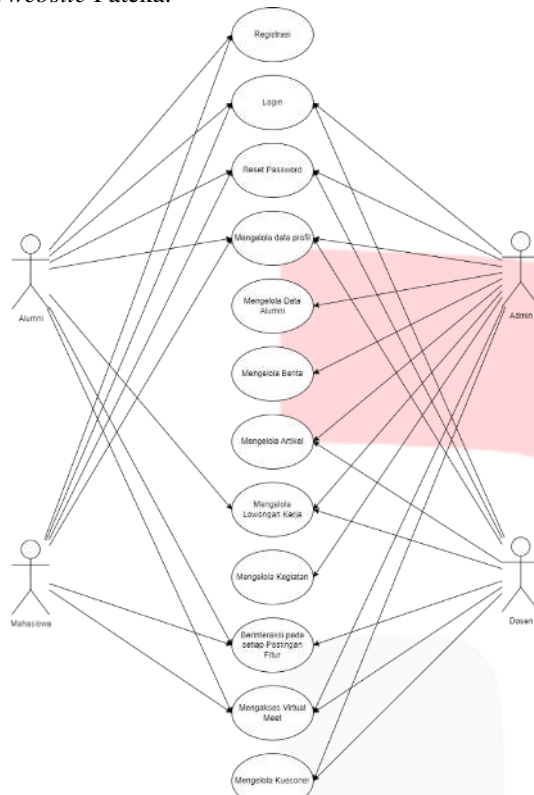
Flowchart user dengan role mahasiswa

Gambar 6 adalah alur pengguna dengan role mahasiswa untuk masuk kedalam dashboard pengguna akan melakukan registrasi terlebih dahulu jika belum memiliki akun lalu pengguna akan masuk ke dalam dashboard dan melakukan

beberapa tahapan-tahapan sesuai dengan *roles* dan *permissions*.

### C. Use Case Diagram

Pada *website* Fateka, perancangan use case diagram memudahkan dalam memodelkan rancangan dan kebutuhan pada *website* Fateka.



GAMBAR 7  
Use Case Diagram Aplikasi Fateka

Pada diagram terdapat empat aktor yang terlibat pada sistem yaitu admin, dosen, alumni, dan mahasiswa.

- UC1: Registrasi
- UC2: Login
- UC3: Reset Password
- UC4: Mengelola Data Profil
- UC5: Mengelola Data Alumni
- UC6: Mengelola Berita
- UC7: Mengelola Artikel
- UC8: Mengelola Lowongan Kerja
- UC9: Mengelola Kegiatan
- UC10: Berinteraksi pada setiap Postingan Fitur
- UC11: Mengakses Virtual Meet
- UC12: Mengelola Kuis

## IV. HASIL DAN ANALISIS

Pengujian *backend* pada aplikasi Fateka dilakukan dengan dua metode pengujian, yaitu: *Unit Testing* menggunakan *framework* laravel dan *library laravel phpunit* untuk melakukan *assertion* pada *object data response* dan *stress testing* menggunakan Apache HTTP server *benchmarking tool* untuk melihat performa *website* dan seberapa banyak user dapat mengakses *request* halaman *website* aplikasi Fateka secara bersamaan.

### A. Hasil Pengujian Unit Testing

Berikut adalah hasil pengujian terhadap fitur aplikasi Fateka. Hasil yang ditampilkan merupakan hasil pengujian menggunakan *framework* laravel dan *library laravel phpunit*.

TABEL 1  
Hasil Unit Testing

| No | Pengujian        | Test Case                                               | Hasil yang diharapkan                               | Hasil Pengujian                         | Keterangan |
|----|------------------|---------------------------------------------------------|-----------------------------------------------------|-----------------------------------------|------------|
| 1  | Registrasi       | Registration screen can be rendered                     | Sistem akan menampilkan halaman registrasi          | Menampilkan halaman registrasi          | Valid      |
| 2  |                  | New users if name is empty                              | Sistem akan mengirimkan pesan error                 | Pesan error terkirim                    | Valid      |
| 3  |                  | New users if email is empty                             | Sistem akan mengirimkan pesan error                 | Pesan error terkirim                    | Valid      |
| 4  |                  | New users if password is empty                          | Sistem akan mengirimkan pesan error                 | Pesan error terkirim                    | Valid      |
| 5  |                  | New users if registered with email that already existed | Sistem akan mengirimkan pesan error                 | Pesan error terkirim                    | Valid      |
| 6  |                  | New users can register                                  | Data input register dapat disimpan ke database      | Data tersimpan                          | Valid      |
| 7  | Login            | Login screen can be rendered                            | Sistem akan menampilkan halaman login               | Menampilkan halaman login               | Valid      |
| 8  |                  | Users can authenticate using the login screen           | Melakukan login                                     | Sukses login                            | Valid      |
| 9  |                  | Users can not authenticate with invalid password        | Sistem akan mengirimkan pesan error                 | Pesan error terkirim                    | Valid      |
| 10 | Password reset   | Reset password link screen can be rendered              | Sistem akan menampilkan halaman link reset password | Menampilkan halaman link reset password | Valid      |
| 11 |                  | Reset password link can be requested                    | Sistem akan mengirimkan link reset password         | Link reset password terkirim            | Valid      |
| 12 |                  | Reset password screen can be rendered                   | Sistem akan menampilkan halaman password reset      | Menampilkan halaman password reset      | Valid      |
| 13 |                  | Password can be reset with valid token                  | Data input password dapat disimpan ke database      | Data tersimpan                          | Valid      |
| 14 | Verifikasi email | Email verification screen can be rendered               | Sistem akan menampilkan halaman verifikasi email    | Menampilkan halaman verifikasi email    | Valid      |

| No | Pengujian           | Test Case                                       | Hasil yang diharapkan                               | Hasil Pengujian                         | Keterangan |
|----|---------------------|-------------------------------------------------|-----------------------------------------------------|-----------------------------------------|------------|
| 15 |                     | Email can be verified                           | Data input email dapat disimpan ke database         | Data tersimpan                          | Valid      |
| 16 |                     | Email is not verified with invalid hash         | Sistem akan mengirimkan pesan error                 | Pesan error terkirim                    | Valid      |
| 17 |                     | Confirm password screen can be rendered         | Sistem akan menampilkan halaman konfirmasi password | Menampilkan halaman konfirmasi password | Valid      |
| 18 | Konfirmasi Password | Password can be confirmed                       | Data input password dapat disimpan ke database      | Data tersimpan                          | Valid      |
| 19 |                     | Password is not confirmed with invalid password | Sistem akan mengirimkan pesan error                 | Pesan error terkirim                    | Valid      |
| 20 | Data Alumni         | Data alumni can be create                       | Data input dapat disimpan ke database               | Data tersimpan                          | Valid      |
| 21 |                     | Data alumni can be update                       | data dapat di update dan disimpan ke database       | Data dapat diupdate                     | Valid      |
| 22 |                     | Data alumni can be deleted                      | Data dapat di hapus dari database                   | Data terhapus                           | Valid      |
| 23 | Berita              | Berita can be create                            | Data input dapat disimpan ke database               | Data tersimpan                          | Valid      |
| 24 |                     | Berita can be update                            | data dapat di update dan disimpan ke database       | Data dapat diupdate                     | Valid      |
| 25 |                     | Berita can be deleted                           | Data dapat di hapus dari database                   | Data terhapus                           | Valid      |
| 26 | Artikel             | Artikel can be create                           | Data input dapat disimpan ke database               | Data tersimpan                          | Valid      |
| 27 |                     | Artikel can be update                           | data dapat di update dan disimpan ke database       | Data dapat diupdate                     | Valid      |
| 28 |                     | Artikel can be deleted                          | Data dapat di hapus dari database                   | Data terhapus                           | Valid      |
| 29 | Lowongan kerja      | Lowongan kerja can be create                    | Data input dapat disimpan ke database               | Data tersimpan                          | Valid      |
| 30 |                     | Lowongan kerja can be update                    | data dapat di update dan disimpan ke database       | Data dapat diupdate                     | Valid      |
| 31 |                     | Lowongan kerja can be deleted                   | Data dapat di hapus dari database                   | Data terhapus                           | Valid      |
| 32 | Kegiatan            | Kegiatan can be create                          | Data input dapat disimpan ke database               | Data tersimpan                          | Valid      |

| No | Pengujian | Test Case               | Hasil yang diharapkan                         | Hasil Pengujian     | Keterangan |
|----|-----------|-------------------------|-----------------------------------------------|---------------------|------------|
| 33 |           | Kegiatan can be update  | data dapat di update dan disimpan ke database | Data dapat diupdate | Valid      |
| 34 |           | Kegiatan can be deleted | Data dapat di hapus dari database             | Data terhapus       | Valid      |

Dari hasil pengamatan pada pengujian setiap fitur, *backend* pada aplikasi Fateka sudah menerima inputan dan juga mengembalikan *response* yang sesuai pada setiap pengujian yang sudah dilakukan terhadap inputan yang valid dan tidak valid pada *request*, berdasarkan hasil pengujian juga diperoleh dari 34 *test case* yang diuji terhadap semua fitur jumlah *test case* yang berhasil dilewati adalah 34 atau 100% berhasil melewati *unit testing*, menunjukkan bahwa *backend* aplikasi Fateka dapat diandalkan untuk memproses dan menyediakan data untuk kebutuhan aplikasi Fateka.

#### B. Hasil Pengujian Stress Testing

Berikut adalah hasil *stress testing*. Hasil yang ditampilkan merupakan hasil pengujian menggunakan Apache Benchmark. Terdapat dua parameter yang digunakan dalam menguji *stress testing*, yaitu:

- n = banyaknya jumlah pengguna yang mengakses ke dalam *server* tujuan
- c = banyaknya jumlah pengguna yang mengakses ke *server* tujuan secara bersamaan

Berikut hasil *stress testing* pada aplikasi Fateka.

TABEL 2  
Hasil Stress Testing

| Skenario Pengujian | Requests (n) | Concurrency (c) | Time taken for tests | Request per second |
|--------------------|--------------|-----------------|----------------------|--------------------|
| 1                  | 50           | 5               | 15.211 seconds       | 3.29/sec           |
| 2                  | 100          | 10              | 29.796 seconds       | 3.36/sec           |
| 3                  | 250          | 25              | 77.618 seconds       | 3.22/sec           |
| 4                  | 500          | 50              | 154.765 seconds      | 3.23/sec           |
| 5                  | 750          | 75              | 236.179 seconds      | 3.18/sec           |
| 6                  | 1000         | 100             | 292.423 seconds      | 3.42/sec           |

Dari hasil pengujian *stress testing* dapat ditarik kesimpulan, *server* aplikasi Fateka dapat melayani sampai dengan 1000 *requests* dengan total rata-rata waktu yang dibutuhkan dibawah 5 detik, dalam hal performa *server* perlu di tingkatkan.

#### V. KESIMPULAN

Berdasarkan hasil tahapan yang telah dilakukan dalam pembuatan *website* aplikasi Fateka dapat ditarik kesimpulan sebagai berikut:

- Membuat *backend* dengan *framework* laravel dapat mempermudah dalam perancangan, karena adanya sistem *Model, View, Controller (MVC)* yang membantu penulis dalam merancang program untuk *backend*. Konsep MVC dapat membuat penulis menjadi mudah memahami alur dari program yang dibuat.
- Setelah melakukan pengujian menggunakan *unit testing*, didapatkan hasil bahwa fungsi yang ada dalam *website*

aplikasi Fateka dapat berjalan dengan baik sesuai dengan ekspektasi dalam *unit testing*. Dan setelah melakukan pengujian menggunakan *stress testing*, penulis mendapatkan hasil menggunakan *apachebench* dengan mengirimkan 1000 *request* ke *server*, dengan hasil yang didapatkan waktu yang dibutuhkan untuk *testing* 292.423 *seconds*, *complete requests* 1000, *concurrency level* 100, dan total rata-rata waktu yang dibutuhkan dibawah 5 detik.

#### REFERENSI

- [1] J. Simarmata, Romindo, S. H. Putra, A. Prasetyo, M. N. H. Siregar, D. P. Yudhi Ardiana, D. Chamidah, B. Purba, Jamaludin, "Teknologi Sistem Informasi," dalam *Teknologi Informasi dan Sistem Informasi Manajemen*, Medan, Yayasan Kita Menulis, 2020, pp. 15-94.
- [2] A. Lutfi, "Sistem Informasi Akademik Madrasah Aliyah Salafiyah Syafi'iyah Menggunakan PHP dan MySQL," *Jurnal Manajemen Informatika dan Komputer*, vol. 3, no. 2, 2017.
- [3] K. Rukun dan B. H. Hayadi, *Sistem Informasi Berbasis Expert System*, Yogyakarta: Deepublish, 2018.
- [4] P. D. Dutonde, S. S. Mamidwar, M. S. Korvate, S. Bafna, Prof. D. D. Shirbhate, "International Journal for Research in Applied Science & Engineering Technology (IJRASET)," *Website Development Technologies: A Review*, vol. X, no. 1, pp. 359-366, 2022.
- [5] R. I. Borman, A. T. Priandika, A. R. Edison, "Implementasi Metode Pengembangan Sistem Extreme Programming (XP)," *Jurnal Sistem dan Teknologi Informasi*, vol. 8, no. 3, pp. 272-277, 2020.
- [6] A. S. Irmawati Carolina, "Penerapan Metode Extreme Programming Dalam Perancangan Aplikasi Perhitungan Kuota SKS Mengajar Dosen," *IKRA-ITH Informatika*, vol. 3, no. 1, pp. 106-113, 2019.
- [7] V. Khorikov, *Unit Testing*, New York: Manning Publications Co., 2020.
- [8] T. Hamilton, "What is STRESS Testing in Software Testing?," *Guru99*, 21 January 2023. [Online]. Available: <https://www.guru99.com/stress-testing-tutorial.html>. [Diakses 17 February 2023].
- [9] R. Juliarto, "Apa itu UML? Beserta Pengertian dan Contohnya - Dicoding Blog," *Dicoding Indonesia*, 12 Mei 2021. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-uml/>. [Diakses 29 Desember 2022].
- [10] M. Ohyver, Jurike V. Moniaga, I. Sungkawa, B. E. Subagyo, I. A. Chandra, "The Comparison Firebase Realtime Database and MySQL Database Performance using Wilcoxon Signed-Rank Test," dalam *Procedia Computer Science*, Jakarta, Procedia Computer Science, 2019, pp. 396-405.