

**PEMBUATAN WEB SERVICE UNTUK WEBSITE PENGGALANGAN
DANA WAKE UP INDONESIA**



Muhammad Nurmansyah Santosa

5235117090

**Skripsi ini Disusun Sebagai Salah Satu Persyaratan untuk Memperoleh
Gelar Sarjana Pendidikan**

**PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER
JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK
UNIVERSITAS NEGERI JAKARTA**

2016

HALAMAN PENGESAHAN

NAMA DOSEN	TANDA TANGAN	TANGGAL
<u>Hamidillah Ajie, S.Si., MT</u> (Dosen Pembimbing I)
<u>M. Ficky Duskarnaen, M.Sc</u> (Dosen Pembimbing II)

PENGESAHAN PANITIA UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
(Ketua Penguji)
(Sekretaris Penguji)
(Dosen Ahli)

Tanggal Lulus:

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis skripsi saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, Januari 2016

Yang membuat pernyataan

Muhammad Nurmansyah Santosa

5235117090

PEMBUATAN WEB SERVICE UNTUK WEBSITE PENGGALANGAN DANA WAKE UP INDONESIA

MUHAMMAD NURMANSYAH SANTOSA

ABSTRAK

Latar belakang dari penelitian ini adalah belum adanya situs Wake Up Indonesia yang menjadi wadah untuk penggalangan dana pada setiap kegiatan yang akan dilakukan oleh pihak yayasan. Untuk memudahkan pengolahan data dan pembuatan situs, situs dipisahkan antara *frontend* dan *backend (service)*.

Tujuan penelitian ini adalah untuk membuat *web service* yang mengolah data dan pengembangan *website* untuk penggalangan dana pada setiap kegiatan yang dilakukan oleh pihak yayasan Wake Up Indonesia. Penelitian ini berfungsi untuk memberikan kemudahan kepada para donatur yang ingin mendonasikan donasinya pada kegiatan yang akan dilakukan oleh pihak yayasan Wake Up Indonesia.

Penelitian dilakukan di Sekretariat Yayasan Wake Up Indonesia dari bulan September 2015 hingga Desember 2015. Metode penelitian yang digunakan adalah *Research and Development*, dan metode pengembangan *software* yang digunakan adalah metode Spiral. Kesimpulan dari penelitian ini adalah terdapat 4 tahap utama yang dilakukan pada pembuatan *web service* ini, yakni (1) Perencanaan (2) Analisis Risiko (3) Rekayasa (4) Evaluasi Pemakai dan pengujian dengan metode pengujian *unit testing*. Berdasarkan hasil uji coba *unit testing*, *web service* ini telah terbukti berfungsi sesuai dengan yang diharapkan.

Kata kunci: Web Service, Metode Spiral, *Unit testing*.

**MANAGEMENT APPLICATION AND NOTIFICATION OF FORMAL
LETTER DISPATCH DOCUMENTATION AT JAKARTA STATE
UNIVERSITY**

MUHAMMAD NURMANSYAH SANTOSA

ABSTRACT

The background of this research is due to the fact that Wake Up Indonesia foundation doesn't have any website yet as a media of fundraising in every activity conducted by the foundation. To ease the data processing and *interface* development, the website development is divided into frontend (interface) and backend (service).

The purpose of this research is to make a web service which process data and website development for fundraising in every activity conducted by Wake Up Indonesia foundation. This research provides ease for donors who wants to donate for the foundation's activity.

The research conducted at the secretariat of Wake Up Indonesia foundation started September 2015 until Desember 2015. The research method used is Research and Development, and the software development method is Spiral. The conclusion of this research is that there are 4 main steps done at the application making process, specifically (1) Plannig (2) Risk Analysis (3) Engineering (4) User evaluation and test using unit testing method. According to unit testing result, this web service is proved to work as expected.

Keywords: Web Service, Spiral Method, Unit testing.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran *Allah* SWT Yang Maha Pengasih dan Maha Penyayang. Atas rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul **Pembuatan Web service Untuk Website Penggalangan Dana Wake Up Indonesia**. Skripsi ini disusun sebagai salah satu syarat dalam memperoleh gelar Sarjana Pendidikan pada Program Studi Pendidikan Teknik Informatika dan Komputer, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.

Selama penyusunan skripsi, banyak bantuan yang penulis terima dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis mengucapkan terima kasih kepada:

1. Ibu Yuliatri Sastrawijaya, M.Pd., selaku Ketua Program Studi Pendidikan Teknik Informatika dan Komputer, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.
2. Bapak Hamidillah Ajie, M.T., selaku Dosen Pembimbing I.
3. Bapak M. Ficky Duskarnaen, M.Sc. selaku Dosen Pembimbing II dan Kepala Pustikom UNJ.
4. Bapak Lipur Sugiyanta, Ph.D. , selaku pembimbing akademik.
5. Seluruh Staf Administrasi Tata Usaha Jurusan Teknik Elektro, Pustikom UNJ, dan Kantor Wakil Rektor 2 Universitas Negeri Jakarta.
6. Bapak Endang Iskandar dan Ibu Sunarsih, selaku kedua orang tua penulis. Terima kasih sudah membesarkan, mendidik, merawat, dan mendoakan aku sehingga dapat menyelesaikan skripsi ini.
7. Seluruh keluarga yang tidak disebutkan satu persatu diskripsi ini. Terima kasih atas doa dan dukungannya.
8. Teman-teman seperjuangan PTIK angkatan 2011, Bang Akbar dan Bang Agung, Rumah Akbar dan Group Facebook PHP Laravel yang telah memberikan motivasi, semangat, dan doanya dalam penulisan skripsi ini.
9. Seluruh pihak yang telah mendukung yang tidak bisa disebutkan

satu persatu demi terselesaikannya skripsi ini dengan baik dan lancar.
Semoga Tuhan membalas semua kebaikan yang telah diberikan.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, karenanya penulis mohon maaf apabila terdapat kekurangan kesalahan baik dari isi maupun tulisan. Akhir kata penulis berharap semoga skripsi ini bermanfaat bagi yang membacanya.

Jakarta, 13 Januari 2016

Muhammad Nurmansyah Santosa

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	
1.1. Latar Belakang Masalah	1
1.2. Identifikasi Masalah	3
1.3. Pembatasan Masalah	3
1.4. Perumusan Masalah	3
1.5. Tujuan Penelitian	3
1.6. Manfaat Penelitian	4
BAB II KERANGKA TEORITIK DAN KERANGKA BERPIKIR	
2.1 Kerangka Teoritik	5
2.1.1. <i>Website</i>	5
2.1.1.1. Pengertian <i>Website</i>	5
2.1.1.2. Perkembangan <i>Website</i>	5
2.1.1.3. Pengembangan Teknologi <i>Website</i>	7
2.1.1.4. Manfaat <i>Website</i>	8
2.1.2. <i>Web Service</i>	9
2.1.2.1 Pengertian <i>Web Service</i>	9
2.1.2.2. <i>Web Service Restful</i>	10
2.1.3. Pengertian <i>Database</i>	12
2.1.3.1. <i>MySQL Database</i>	13

2.1.3.2. Analisis dan Perancangan <i>Database</i>	18
2.1.3.3. JSON	20
2.1.3.4. PHP	21
2.1.4. <i>Framework</i>	22
2.1.4.1. Pengertian <i>Framework</i>	22
2.1.4.2. Laravel.....	23
2.1.5. API Doc	26
2.1.6. Postman	26
2.1.7. Wake Up Indonesia	26
2.1.7.1. Profil Wake Up Indonesia.....	26
2.1.7.2. Program Kerja	27
2.1.7.3. Penggalangan Dana	30
2.1.8. Pengembangan <i>Web Service</i> Menggunakan Metode Spiral	30
2.2 Kerangka Berpikir	33

BAB III METODELOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian	36
3.2. Prosedur Penelitian	36
3.3. Metode Penelitian	37
3.4. Perencanaan dan Perancangan <i>Web Service</i>	37
3.4.1. Perencanaan (<i>Planning</i>).....	37
3.4.1.1. Pengumpulan Data	37
3.4.1.2. Observasi	38
3.4.1.3. Wawancara	38
3.4.1.4. Rancangan Kebutuhan Fungsional	38
3.4.1.5. Rancangan <i>Database</i>	39
3.4.2. Analisis Kebutuhan dan Risiko.....	42
3.4.2.1. Kriteria Keberhasilan Sistem	46
3.5. Tipe <i>Request</i>	46
3.5.1. <i>Get</i>	46
3.5.2. <i>Post</i>	47
3.5.3. <i>Put</i>	47

3.6. Hasil Tipe <i>Request</i>	47
3.6.1. Method <i>Get</i>	47
3.6.2. Method <i>Post</i>	48
3.6.3. Method <i>Put</i>	48
3.7. Rekayasa Pengembangan.....	48
3.7.1. Rekayasa Pengembangan Sistem	48
3.7.2. Evaluasi	49
BAB IV HASIL PENELITIAN	
4.1. Hasil Penelitian	57
4.1.1. Hasil Pengujian Fungsional	57
4.1.1.1. Hasil Pengujian Fungsional Method <i>Get</i>	57
4.1.1.2. Hasil Pengujian Fungsional Method <i>Post</i>	71
4.1.1.3. Hasil Pengujian Fungsional Method <i>Put</i>	75
4.2. Pembahasan	77
4.2.1. Pembahasan Hasil Pengujian Fungsional	77
BAB V KESIMPULAN, DAN SARAN	
5.1. Kesimpulan	78
5.1.1. Implikasi di Bidang Keteknikan	78
5.1.2. Implikasi di Bidang Pendidikan	79
5.2. Saran	79
DAFTAR PUSTAKA	80
LAMPIRAN.....	81
DAFTAR RIWAYAT HIDUP	101

DAFTAR TABEL

	Halaman
Tabel 3.1 Data Tabel <i>User Non Member</i>	38
Tabel 3.2 Data Tabel <i>User Member</i>	39
Tabel 3.3 Data Tabel Admin	39
Tabel 3.4 Data Tabel Donasi	40
Tabel 3.5 Data Tabel Kategori	40
Tabel 3.6 Data Tabel Pengguna	40
Tabel 3.7 Data Tabel <i>Project</i>	41
Tabel 3.8 Data Tabel Relawan	41
Tabel 3.9 Data Tabel <i>Users</i>	41
Tabel 3.10 Daftar Kebutuhan Fungsional Sistem dan Analisis Risiko	42
Tabel 3.11 Daftar <i>Endpoint</i> Berdasarkan Kebutuhan Fungsional Dengan Menggunakan URL http://localhost/wui-service/api/ ..	44
Tabel 3.12 Daftar <i>Endpoint, Method, Deskripsi, dan Parameter</i> Dengan Menggunakan URL http://localhost/wui-service/api/	45
Tabel 3.13 Data Kriteria Keberhasilan <i>Web Service</i>	46
Tabel 3.14 Data Tabel Kebutuhan Fungsional Tipe <i>Request Get</i>	47
Tabel 3.15 Data Tabel Kebutuhan Fungsional Tipe <i>Request Post</i>	47
Tabel 3.16 Data Tabel Kebutuhan Fungsional Tipe <i>Request Put</i>	47
Tabel 3.17. Data Tabel Hasil <i>Endpoint Method Get</i>	47
Tabel 3.18. Data Tabel Hasil <i>Endpoint Method Post</i>	48
Tabel 3.19. Data Tabel Hasil <i>Endpoint Method Put</i>	48
Tabel 3.20. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Get</i> Menampilkan Seluruh <i>Project</i> Secara Partial	50
Tabel 3.21. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Get</i> Mengirim Kode Verifikasi Akun	50
Tabel 3.22. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Get</i> Menampilkan Profil <i>User</i>	50
Tabel 3.23. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Get</i> Menampilkan Record Donasi <i>User</i>	51

Tabel 3.24. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Get</i> Melakukan Pencarian <i>Project</i> Berdasarkan <i>Keyword</i>	51
Tabel 3.25. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Get</i> Menampilkan <i>Project</i> Berdasarkan Kategori Secara Partial	52
Tabel 3.26. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Get</i> Menampilkan Data Sebuah <i>Project</i>	52
Tabel 3.27. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Get</i> Melakukan Pengecekan Apakah Masih Dapat Melakukan Donasi Pada Sebuah <i>Project</i>	52
Tabel 3.28. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Post</i> Mengajukan Donasi Untuk Sebuah <i>Project</i>	53
Tabel 3.29. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Post</i> Mengajukan Donasi Untuk Sebuah <i>Project</i> (<i>Non Member</i>)	53
Tabel 3.30. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Post</i> Melakukan Cek Terhadap Status Donasi	53
Tabel 3.31. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Post</i> Mengajukan Diri Sebagai Relawan Dalam <i>Project</i>	54
Tabel 3.32. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Post</i> Otentikasi <i>User</i>	54
Tabel 3.33. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Post</i> Mendaftarkan <i>User</i>	54
Tabel 3.34. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Post</i> Mengirim Ulang Email Verifikasi Akun <i>User</i>	55
Tabel 3.35. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Post</i> Melakukan Reset Password.....	55
Tabel 3.36. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Put Update</i> Nama Depan dan Nama Belakang <i>User</i>	55
Tabel 3.37. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i> <i>Put</i> Ganti Password <i>User</i>	56
Tabel 3.38. Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request</i>	

	<i>Put Upload Foto User</i>	56
Tabel 4.1.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Get</i> Menampilkan Seluruh <i>Project</i> Secara Partial	57
Tabel 4.2.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Get</i> Mengirim Kode Verifikasi Akun	59
Tabel 4.3.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Get</i> Menampilkan Profil <i>User</i>	60
Tabel 4.4.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Get</i> Menampilkan Record Donasi <i>User</i>	62
Tabel 4.5.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Get</i> Melakukan Pencarian <i>Project</i> Berdasarkan <i>Keyword</i>	65
Tabel 4.6.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Get</i> Menampilkan <i>Project</i> Berdasarkan Kategori Secara Partial	67
Tabel 4.7.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Get</i> Menampilkan Data Sebuah <i>Project</i>	68
Tabel 4.8.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Get</i> Melakukan Pengecekan Apakah Masih Dapat Melakukan Donasi Pada Sebuah <i>Project</i>	70
Tabel 4.9.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Post</i> Mengajukan Donasi Untuk Sebuah <i>Project</i>	71
Tabel 4.10.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Post</i> Mengajukan Donasi Untuk Sebuah <i>Project (Non Member)</i>	71
Tabel 4.11.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Post</i> Melakukan Cek Terhadap Status Donasi	72
Tabel 4.12.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Post</i> Mengajukan Diri Sebagai Relawan Dalam <i>Project</i>	73
Tabel 4.13.	Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Post</i> Otentikasi <i>User</i>	73

Tabel 4.14. Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Post</i> Mendaftarkan <i>User</i>	74
Tabel 4.15. Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Post</i> Mengirim Ulang Email Verifikasi Akun <i>User</i>	74
Tabel 4.16. Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Post</i> Melakukan Reset Password	75
Tabel 4.17. Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Put Update</i> Nama Depan dan Nama Belakang <i>User</i> ..	75
Tabel 4.18. Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Put</i> Ganti Password <i>User</i>	76
Tabel 4.19. Hasil Pengujian Fungsional Pada <i>Endpoint</i> Dengan Tipe <i>Request Put</i> Upload Foto <i>User</i>	76

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Sistem <i>Web Service</i> RestFul.....	12
Gambar 2.2. Survey Sitepoint.com	24
Gambar 2.3. Aktivitas Metode Spiral	32
Gambar 2.4. Kerangka Berpikir	35
Gambar 3.1. ERD <i>Database</i> Website Penggalangan Dana Wake Up Indonesia.....	42

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Menampilkan Surat Bahwa Penulis Telah Melakukan Penelitian di Wake Up Indonesia	81
Lampiran 2. Menampilkan Seluruh <i>Project</i> Secara Partial	82
Lampiran 3. Menampilkan Profil <i>User</i>	83
Lampiran 4. Menampilkan Record Donasi <i>User</i>	84
Lampiran 5. Melakukan Pencarian <i>Project</i> Berdasarkan <i>Keyword</i>	85
Lampiran 6. Menampilkan <i>Project</i> Berdasarkan Kategori Secara Partia ...	86
Lampiran 7. Menampilkan Data Sebuah <i>Project</i>	87
Lampiran 8. Melakukan Pengecekan Apakah Masih Dapat Melakukan Donasi Pada Sebuah <i>Project</i>	88
Lampiran 9. Mengajukan Donasi Untuk Sebuah <i>Project</i>	89
Lampiran 10. Mengajukan Donasi Untuk Sebuah <i>Project (Non Member)</i> ..	90
Lampiran 11. Melakukan Cek Terhadap Status Donasi	91
Lampiran 12. Mengajukan Diri Sebagai Relawan Dalam <i>Project</i>	92
Lampiran 13. Otentikasi <i>User</i>	93
Lampiran 14. Mendaftarkan <i>User</i>	94
Lampiran 15. Mengirim Ulang Email Verifikasi Akun <i>User</i>	95
Lampiran 16. Melakukan <i>Reset Password</i>	96
Lampiran 17. <i>Update</i> Nama Depan dan Nama Belakang	97
Lampiran 18. Ganti Password <i>User</i>	98
Lampiran 19. Upload Foto <i>User</i>	99

BAB I

PENDAHULUAN

1.1. Latar Belakang

Yayasan Wake Up Indonesia adalah sebuah gerakan pemuda yang memiliki Visi untuk membangun Indonesia melalui kontribusi nyata yang mereka dapat kerjakan. Gerakan Wake Up Indonesia memiliki program-program meliputi berbagai bidang di antaranya edukasi, lingkungan, olahraga, seni budaya, dan sosial. Gerakan ini bersifat sosial, memberikan penyadaran kepada pemuda agar turut serta dalam membangun negeri ini melalui hal-hal sederhana yang mereka bisa, berupa donasi dalam bentuk materi ataupun tenaga.

Yayasan Wake Up Indonesia bergerak dibidang sosial, untuk pembiayaan program kerja dan kegiatan yang dilakukan oleh pihak yayasan menggunakan dana yang diperoleh dari pengiriman proposal ke lembaga atau instansi *benefit*. Selain itu pihak yayasan melakukan kegiatan-kegiatan produktif seperti mengadakan pameran dan bazar yang keuntungannya dimasukkan kedalam kas yayasan.

Di era yang sudah modern ini, penggunaan media terutama perkembangan *web* sangat berpengaruh dalam berbagai aspek, salah satunya adalah untuk penggalangan dana untuk yayasan Wake Up Indonesia. Wake Up Indonesia belum mempunyai *website* sebagai sarana untuk membantu mengumpulkan dana. Pengumpulan dana secara *online*, dengan memanfaatkan *website* akan lebih efisien dari segi waktu dan tenaga.

Untuk mengatasi permasalahan tersebut, telah direncanakan pembuatan *website* penggalangan dana yayasan Wake Up Indonesia. Sistem yang ditawarkan

berupa pemberian donasi dari setiap kegiatan yang akan diselenggarakan. Agar mempermudah pemilihan dalam pengerjaan, *website* yang digunakan akan dibagi menjadi dua sisi, yaitu *frontend* dan *backend*. *Frontend* merupakan *client side* yang menampilkan *view* yang akan dilihat oleh *user* sedangkan *backend* atau sering disebut dengan *service/webservice* merupakan *server side* yang mengolah data dan menjalankan *business logic website*.

Web service merupakan suatu sistem yang dirancang untuk mendukung komunikasi antar sistem (aplikasi) dalam suatu jaringan, artinya didalam jaringan yang sama *web service* dapat diakses oleh aplikasi lain (*frontend*) melalui *web*. Komunikasi data melalui *web service* dilakukan melalui *http* atau *Internet protocol*.

Web service hanya menyediakan *business logic*, proses, dan data antar jaringan. *Web service* memungkinkan sebuah aplikasi menjadi sangat kecil ukurannya, karena datanya disimpan di *web service* sehingga tidak perlu disimpan secara lokal pada *website*. *Web service* ini juga memudahkan untuk memperbaharui data dalam *website* karena perubahan cukup dilakukan di *web service* dan semua aplikasi yang terinstall secara lokal dan mengakses *web service* ini pun akan secara otomatis mengikuti perubahan yang dibuat.

Web Service itu sendiri bermanfaat untuk pengembangan *website Wake Up Indonesia*, karena *web service* dapat di implementasikan terhadap *Cross Platform* lainnya.

Pada penelitian ini, penulis lebih menitik beratkan kepada pembuatan *service* atau *backend* untuk mengolah data pada *website Wake Up Indoneisa* dan

menuliskanmya dalam laporan skripsi dengan judul “Pembuatan *Web service* untuk *website* Penggalangan Dana Wake Up Indonesia”.

1.2. Identifikasi Masalah

1. Tidak adanya *website* penggalangan dana Wake Up Indonesia.
2. Tidak adanya *web service* untuk pengolah data dan pengembangan *website* penggalangan dana Wake Up Indonesia.

1.3. Pembatasan Masalah

Berikut batasan-batasan masalah dalam penelitian ini:

1. *Website* ini dibangun dengan menggunakan PHP sebagai *Server side Programming* dan *MySQL* sebagai *database server*nya.
2. Dibatasi pembagian *website*, peneliti hanya membuat *web service* dan *database*.

1.4. Rumusan Masalah

Berikut adalah permasalahan yang akan dibahas dalam penelitian ini:

“Bagaimana Pengembangan *Web Service Website* Penggalangan Dana Untuk Yayasan Wake Up Indonesia?”

1.5 Tujuan Penelitian

Tujuan umum dari penelitian ini adalah membuat *web service* sebagai pengolah data dan pengembangan *website* penggalangan dana untuk Yayasan Wake Up Indonesia.

1.6 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat berguna bagi beberapa pihak yang terkait, yakni:

1. Bagi penulis, penelitian ini diharapkan mampu menambah wawasan dan pengetahuan yang berkenaan dengan proses pembuatan *web service*.
2. Bagi pihak Yayasan, penelitian ini diharapkan mampu memberikan kemudahan kepada Yayasan untuk mengolah data dari penggalangan dana secara *online*.
3. Bagi peneliti lain, penelitian ini mampu menambah menjadi referensi, acuan, dan juga koreksi jika ingin mengadakan penelitian sejenis.

BAB II

KERANGKA TEORETIK DAN KERANGKA BERPIKIR

2.1. Kerangka Teoretik

2.1.1. *Website*

2.1.1.1. Pengertian *Website*

Menurut Abdul Kadir (2003:374) *website* atau *website* merupakan kumpulan beberapa halaman pribadi, organisasi, atau perusahaan yang menampilkan dan memuat informasi dalam *web server* (suatu unit komputer yang berfungsi untuk menyimpan informasi dan mengelola jaringan komputer). Informasi-informasi dalam *website* biasanya dilihat secara *online*. *Website* penggalangan dana Wake Up Indonesia, menampilkan informasi-informasi yang berkenaan dengan tata cara melakukan sumbangan secara *online*.

2.1.1.2. Perkembangan *Website*

Menurut Tim (1994) Perkembangan *website* berdasarkan teknologi dan cara penggunaannya adalah sebagai berikut:

1. *Web 1.0*

Web 1.0 merupakan teknologi awal dari *website*, dimana pembuat sebagai pemberi informasi dan pengguna hanya sebagai pembaca. Bahasa yang digunakan pada *web* ini masih berupa HTML saja.

2. *Web 2.0*

Web 2.0 muncul sekitar tahun 2003 atau 2004, dimana para pengguna *website*-pun dapat berkomunikasi 2 arah dan memiliki berbagai kelebihan lainnya.

Kelebihan dari *web* ini adalah sebagai berikut menurut O'Reilly media: *The Web as Platform* "Pengerjaan suatu aplikasi/tulisan dapat langsung dikerjakan di media internet tanpa harus mengerjakannya terlebih dahulu di windows desktop" *Harnessing Collective Intelligence* "Web 2.0 memiliki kinerja untuk memanfaatkan tulisan orang lain untuk mengisi konten *web* secara kolektif (tidak hanya *webmaster* yang mengisi konten sendiri)". *Data is the Next Intel Inside* "merupakan suatu garansi kepercayaan dari para pemberi data kepada pemilik *website* bahwa pada era *web* 2.0 data sangatlah penting dan harus di *update* setiap waktu" *End of the Software Release Cycle* "pada *web* 2.0 aplikasi *software* dapat langsung digunakan lewat internet/internet menjadi *platform* menjalankan program" *Lightweight Programming Models* "pembuatan *web* 2.0 menggunakan bahasa yang ringan dan mendukung pengembangan program".

3. *Web* 3.0

Web ini diperkirakan akan berkembang pada tahun 2010 sampai 2020 dan saat ini masih dalam tahap pengembangan. Menurut PC magazine karakteristik dari *web* 3.0 adalah : *Semantic Web* "*web* dengan kemampuan membaca *website* semudah manusia membacanya sehingga informasi dapat disajikan dengan cepat dan tepat" *The 3D Web* (*web* dengan kemampuan visual 3D dan interaksi secara *realtime*) *The Media-Centric Web* (Photo, audio, dan video akan menjadi cara lain untuk mencari informasi yang kita inginkan selain *keyword*) *The Pervasive Web* (*Web* yang mudah diakses dengan berbagai cara dan alat berbeda kapan saja dan dimana saja)

2.1.1.3. Pengembangan Teknologi *Website*.

1. *Frontend*

“*Frontend*” mengacu pada setiap aspek dari proses desain yang muncul dalam atau berhubungan langsung dengan *browser* yang berkaitan dengan desain antarmuka *website*. Tugas-tugas berikut umumnya dianggap sebagai tugas *frontend*:

- a. Desain grafis dan pembuatan gambar
- b. Desain antarmuka
- c. Desain Informasi seperti berkaitan dengan pengalaman pengguna dari *websitenya*
- d. Dokumen HTML dan pengembangan *stylesheet* dengan CSS
- e. *JavaScript* dan *jQuery*

2. *Backend*

“*Backend*” mengacu pada program dan skrip yang bekerja pada *server* di balik layar untuk membuat halaman *web* yang dinamis dan interaktif. Secara umum, pengembangan *backend web* berada di tangan *programmer* berpengalaman, tapi itu baik untuk semua desainer *web* agar terbiasa dengan fungsi *backend*.

Tugas-tugas berikut yang umumnya berlangsung pada *backend*:

- a. Desain informasi seperti berkaitan bagaimana informasi diatur di *server*
- b. Pemrosesan *form*
- c. Pemrograman *database*
- d. Content management systems

- e. Aplikasi berbasis *web server* lain menggunakan PHP, JSP, Ruby, ASP.NET, Java, dan bahasa pemrograman lainnya

Dari tugas-tugas yang disebutkan di atas *bckend* biasanya terdiri dari tiga bagian: *server*, aplikasi, dan *database*.

Teknologi ini dibangun bekerja sama dengan pengembang *backend*. Teknologi *backend* biasanya terdiri dari bahasa seperti PHP, JSP, Ruby, ASP.NET, Java, dan bahasa pemrograman lainnya Untuk membuat mereka lebih mudah untuk menggunakan mereka biasanya ditingkatkan dengan kerangka seperti Ruby on Rails, CakePHP, dan CodeIgniter yang semua membuat pengembangan lebih cepat dan lebih mudah untuk berkolaborasi

2.1.1.4. Manfaat Website

Adapun manfaat *website* sebagai berikut:

1. Memperluas jangkauan promosi.
2. Media tanpa batas.
3. Internet bisa diakses oleh seluruh lapisan masyarakat
4. Promosi terluas, internet adalah media promosi terluas jika dilihat dari kacamata jangkauan atau cakupan area (*unlimited scopt of areas*).
5. Media pengenalan perusahaan.
6. Sebagai media promosi, Internet merupakan salah satu media promosi yang menarkan biaya yang relatif efisien dikaitkan dengan jangkauan area dengan optimalisasi manfaat terbesar.

7. Meningkatkan *image* bisnis. Tampilan desain *web* yang baik adalah cara yang baik untuk menanamkan kepercayaan dimata pelanggan/calon pelanggan dan dapat meningkatkan penjualan dan keuntungan.
8. Meningkatkan layanan pelanggan.

2.1.2. *Web service*

2.1.2.1. Pengertian *Web service*

Web service adalah aplikasi sekumpulan data (*database*) atau perangkat lunak (*software*) yang dapat diakses secara *remote* oleh berbagai cross platform lainnya dengan sebuah perantara tertentu. Secara umum, *web service* dapat diidentifikasi dengan menggunakan URL seperti halnya *web* pada umumnya. Namun yang membedakan *web service* dengan *web* pada umumnya adalah interaksi yang diberikan oleh *web service*. Berbeda dengan URL *web* pada umumnya, URL *web service* hanya mengandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi. Menurut Wahli, dkk (2006: 7) *web service* merupakan suatu komponen software yang merupakan selfcontaining, aplikasi modular self-describing yang dapat dipublikasikan, dialokasikan, dan dilaksanakan pada web.

Menurut W3C (2004) pengertian *Web service* adalah sebuah sistem *software* yang di desain untuk mendukung *interoperabilitas* interaksi mesin ke mesin melalui sebuah jaringan. *Interface web service* dideskripsikan dengan menggunakan *format* yang mampu diproses oleh mesin (khususnya *WSDL*). Sistem lain yang akan berinteraksi dengan *web service* hanya memerlukan *SOAP*,

yang biasanya disampaikan dengan HTTP dan XML sehingga mempunyai korelasi dengan standar *Web*.

Web service dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah *database* ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan di *platform* apa sebuah data itu dikonsumsi. *Web service* mampu menunjang interoperabilitas. Sehingga *web service* mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada.

Web pada umumnya digunakan untuk melakukan *respon* dan *request* yang dilakukan antara *client* dan *server*. Sebagai contoh, seorang pengguna layanan *web* tertentu mengetikkan alamat url *web* untuk membentuk sebuah *request*. *Request* akan sampai pada *server*, diolah dan kemudian disajikan dalam bentuk sebuah *respon*. Dengan singkat kata terjadilah hubungan *client-server* secara sederhana.

Sedangkan pada *web service* hubungan antara *client* dan *server* tidak terjadi secara langsung. Hubungan antara *client* dan *server* dijembatani oleh file *web service* dalam format tertentu. Sehingga akses terhadap *database* akan ditanggapi tidak secara langsung oleh *server*, melainkan melalui perantara yang disebut sebagai *web service*. Peran dari *web service* ini akan mempermudah distribusi sekaligus integrasi *database* yang tersebar di beberapa *server* sekaligus.

2.1.2.2. *Web service Restful*

RESTful Web service atau juga dikenal dengan nama *RESTful Web API* merupakan jenis *web service* yang di implementasikan dengan menggunakan *link-*

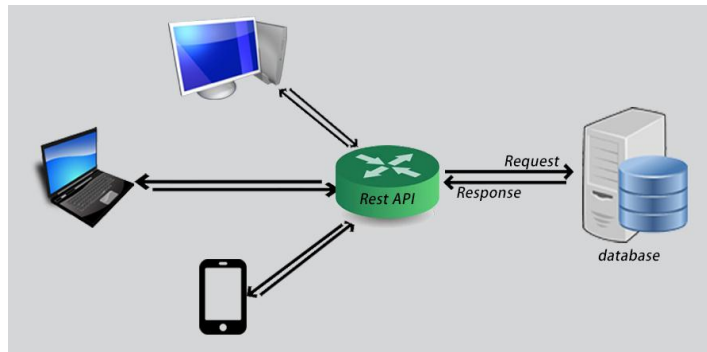
link HTTP dengan menggunakan prinsip-prinsip *REST*. *Service* yang digunakan oleh *web client* menggunakan *method* milik HTTP antara lain *GET*, *PUT*, *POST*, dan *DELETE* untuk berkomunikasi dengan *web service* dan balasan yang dikirimkan adalah dalam bentuk *response* data yang *request*.

Menurut Oliviera, dkk, (2010: 20) *REST* (Representational State Transfer) merupakan arsitektur umum yang digunakan dalam pembangunan *Web service* karena didasarkan pada HTTP untuk berkomunikasi melalui jaringan yang didukung oleh semua *smartphone*. Arsitektur *REST* didasarkan pada komunikasi *client-server*, dimana *client* meminta sekumpulan data yang tersedia dari *web service*.

Untuk penggunaan *method* *GET*, *POST*, *PUT*, atau *DELETE* bisa sesuai kebutuhan.

1. *GET*: Dapat digunakan apabila *request* ke *server* merupakan data yang sedikit. Karena pada *GET* terdapat keterbatasan data. Contohnya adalah penggunaan URL di *browser*.
2. *POST*: Dapat digunakan apabila pengiriman data dengan data yang banyak. Contohnya adalah data yang ada di *form* dengan *method post*.
3. *PUT*: Dapat digunakan untuk pengiriman *file* melalui *service*. Contohnya adalah *upload file*.
4. *DELETE*: *Method* ini sangat jarang digunakan karena terkadang dapat diwakili oleh *GET*. Akan tetapi, ada baiknya digunakan untuk proses penghapusan data di *server*.

Proses pada *sistem web service RestFul* akan diilustrasikan pada Gambar 2.1.



Gambar 2.1 Sistem *Web service RestFul*

2.1.3. Pengertian *Database*

Database adalah kumpulan data yang secara logik berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu (Hariyanto, Bambang, 2004: 20). *Database* adalah mekanisme yang digunakan untuk menyimpan informasi atau data (Stephen dan Plew, 2000: 31). Dengan *database*, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil, dimodifikasi, dihapus, dan ditambah. Cara data disimpan dalam *database* menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria.

Pada penelitian ini, *database* dibuat dan dirancang menggunakan *tools php MyAdmin* dengan *Structur Query Language (SQL)*. SQL adalah bahasa yang digunakan untuk mengelola data pada *Relational Database Management System (RDBMS)*. Secara umum, SQL terdiri dari dua bahasa, yaitu *Data Definition Language (DDL)* dan *Data Manipulation Language (DML)*. Implementasi DDL dan DML berbeda untuk setiap sistem manajemen *database*.

2.1.3.1. MySQL Database

MySQL adalah sebuah *software* sistem untuk manajemen bisnis data SQL atau DBMS (*Database Management System*) yang *multithread*, *multi user*. MySQL adalah sebuah implementasi dari sistem manajemen *database* relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan *software* tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL pertama kali dibuat dan dikembangkan di Swedia, yaitu oleh David Axmark, Allan Larsson dan Michael “Monty” Widenius. Mereka mengembangkan MySQL sejak tahun 1980-an. Saat ini versi MySQL yang sudah stabil mencapai versi 5x, dan sedang dikembangkan versi 6x. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Menurut M.Rudyanto Arief (2011:73) Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. MySQL mendukung operasi *database* transaksional maupun operasi *database* non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan *software server database competitor* lainnya. Namun pada modus

non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis *web* (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus *database* transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional. Menurut M. Rudyanto Arief (2011: 73) Beberapa keistimewaan MySQL sehingga sering dipakai untuk *website*, antara lain :

1. *Portabilitas*. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X *Server*, Solaris, Amiga, dan masih banyak lagi.
2. *Software* sumber terbuka. MySQL didistribusikan sebagai *software* sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*. MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed/unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

6. Perintah dan fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani *database* dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu table serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Skalabilitas dan Pembatasan. MySQL mampu menangani *database* dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu table serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
10. Lokalisasi, MySQL dapat mendeteksi kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya
11. Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tools*) yang dapat digunakan untuk administrasi *database*, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan *database* lainnya semacam PostgreSQL ataupun Oracle.
14. MySQL tersedia di berbagai *platform* dan kompatibel dalam berbagai sistem operasi.
15. Fitur-fitur yang dimiliki MySQL merupakan fitur-fitur yang biasanya banyak dibutuhkan dalam aplikasi *web*.
16. MySQL memiliki overhead koneksi yang rendah, sehingga kecepatan koneksi relatif tinggi.

Menurut M. Rudyanto Arief (2011: 76) kelebihan MySQL :

1. Berlisensi GPL dan *Multi Platform*.
2. Dapat diintegrasikan dengan beberapa bahasa pemrograman seperti .Net, Java, Python, Perl yang merupakan bahasa pemrograman yang paling dominan di kalangan *programmer*.
3. Mendukung ODBC untuk sistem operasi Windows sehingga bisa digunakan aplikasi yang berjalan di windows.
4. Bisa dijalankan pada spesifikasi hardware yang rendah karena lebih hemat *resource memory* (dibandingkan *database* lain) sehingga mudah digunakan untuk bahan pembelajaran.
5. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari 20 bahasa meskipun bahasa Indonesia belum termasuk didalamnya.

Menurut M. Rudyanto Arief (2011: 77) Kekurangan MySQL :

1. Banyak mengklaim kurang *support* terhadap pemrograman Visual/Desktop, sehingga sedikit yang menggunakan untuk aplikasi visual.
2. Karena berlisensi GPL sehingga sulit mendapatkan *update* untuk *problem urgent*, sehingga perusahaan skala menengah keatas lebih memilih RDBMS berlisensi dan di *support* seperti Oracle dan MS SQL Server.
3. Sangat diragukan dalam menangani data skala besar, karena ada beberapa opini yang pro dan kontra terhadap kemampuan MySQL terhadap pengolahan data yang besar.

Menurut M. Rudyanto Arief (2011: 78) Kemampuan MySQL :

1. Memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
2. Memiliki lebih banyak type data seperti : *signed/unsigned integer* yang memiliki panjang data sebesar 1,2,3,4 dan 8 byte, *FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET dan tipe ENUM*.
3. Mendukung *field* yang dijadikan *index*, dengan maksimal 32 *index* dalam satu tabel.
4. MYSQL memiliki beberapa lapisan keamanan, seperti *subnetmask*, nama *host*, dan izin akses *user* dengan system perizinan yang mendetail serta sandi/password terenkripsi.
5. Konektivitas, MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

6. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
7. *Command and function*, MySQL memiliki fungsi dan operator secara penuh yang mendukung perintah *select* dan *where* dalam *query*.
8. *Structure Table*, MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE* dibandingkan DBMS lainnya.
9. Mendukung penuh terhadap kalimat *SQL GROUP BY* dan *ORDER BY*. Mendukung terhadap fungsi penuh (*COUNT()*, *DISTINCT()*, *AVG()*, *STD()*, *SUM()*, *MAX()* dan *MIN()*).

2.1.3.2. Analisis dan Perancangan Database

Menurut Ramon A dan Pauline K (2007:1) Proses analisis dan perancangan *database* dilakukan guna mendefinisikan struktur objek yang ada dalam sistem klien atau *server*. Beberapa prinsip yang perlu diperhatikan dalam perancangan *database* yakni:

1. Kecepatan dan Kemudahan, data harus mampu diakses dan dimanipulasi secara cepat dan mudah.
2. Efisiensi ruang penyimpanan, data harus terbebas dari pengulangan (*redundancy*) sehingga ruang penyimpanan tidak sampai membesar.
3. Keakuratan, data harus bersifat unik guna menekan tingkat ketidakakuran yang mungkin dapat terjadi karena sifat *anomaly* yang mungkin ada pada data.
4. Ketersediaan, data harus mampu diakses dari berbagai lokasi dengan baik.
5. Kelengkapan, data harus lengkap dan terdefinisi dengan baik.

6. Keamanan, data harus dipastikan aman dari berbagai macam kemungkinan serangan atau gangguan.
7. Kebersamaan, data harus dapat diakses oleh berbagai pengguna dalam waktu yang bersamaan dengan baik.

Tahapan perancangan *database* secara umum terdiri dari tiga fase antara lain (Kadir, Abdul, 2008:24-28):

1. Perancangan Konseptual

Pada perancangan konseptual data yang dibutuhkan oleh organisasi/perusahaan dikelompokkan menurut kriteria tertentu. Kemudian antara satu grup data dengan grup data yang lain dilengkapi dengan hubungan. Dalam terminologi *database*, grup data tersebut dinamaka entitas. Adapun hubungan antarentitas biasa dijabarkan dengan menggunakan diagram E-R (*Entity-Relationship*).

2. Perancangan Logis

Perancangan logis merupakan suatu tahapan yang digunakan untuk menentukan hasil perancangan konseptual ke dalam bentuk yang nantinya akan diimplementasikan dalam DBMS.

3. Perancangan Fisik

Pada perancangan fisik sangat dispesifikan terhadap DBMS yang digunakan. Sebagai contoh, tipe data atau domain untuk masingmasing kolom dalam setiap tabel harus disesuaikan dengan DBMS yang digunakan.

2.1.3.3. JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data komputer yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat oleh komputer. JSON menggunakan bahasa *pemrograman* keluarga C, seperti C, C++, Java, Python, dll. JSON terbuat dari dua struktur data universal yang pada dasarnya semua bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini. JSON menggunakan bentuk sebagai berikut:

1. *Object*

Object adalah sepasang nama atau nilai yang tidak terurutkan. Cara penulisan *Object* dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan: (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).

2. *Array*

Array adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).

3. *Value*

Nilai (*value*) dapat berupa sebuah *string* dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.

4. *String*

String adalah kumpulan dari nol atau lebih karakter *Unicode* yang dibungkus dengan tanda kutip ganda. Di dalam *string* dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string*. *String* sangat mirip dengan *string* C atau *Java*.

5. *Number*

Angka (*Number*) sangat mirip dengan angka di C atau *Java*, kecuali format oktal dan heksadesimal tidak digunakan.

Dalam pengembangan aplikasi *web* maupun *mobile* terdapat otentifikasi yang merupakan bagian paling penting dan rumit yang menuntut pendekatan yang lebih baik. Solusi terbaik dalam otentifikasi masalah untuk API adalah JSON *Web Token* (*JWT*).

Menurut Sandro Pasquali (2015:126) JSON *Web Token* merupakan standar JSON yang digunakan untuk mengirimkan informasi yang dapat diverifikasi dan dipercaya dengan cara tanda tangan digital. Ini terdiri dari persetujuan dan objek URL JSON, yang mana tanda tangan kriptografi untuk memverifikasi keaslian. JSON *Web Token* menggunakan bahasa pemrograman yang berbeda, yaitu: *Java*, *PHP*, *Python*, *Node.js*, dan *JavaScript*.

2.1.3.4. PHP Hypertext Processing (PHP)

Menurut Edy Winarno (2011: 53) PHP adalah bahasa pemrograman untuk mengembangkan *website* yang disisipkan pada script HTML. Bahasa ini mempunyai kelebihan yaitu kompatibilitasnya dengan berbagai macam jenis

database dan dukungan dengan berbagai macam jenis sistem operasi. PHP lebih cocok dan umum digunakan jika digabungkan dengan *database* MySQL.

Secara khusus, PHP dirancang untuk membentuk *website* yang dinamis dengan fungsi CRUD (*Create, Read, Update, Delete*) sesuai kebutuhan *user*. Oleh karena kelebihan tersebut, dalam pengimplementasian fungsi-fungsi untuk mengembangkan *website* penggalangan dana Wake Up Indonesia menggunakan bahasa pemrograman PHP.

2.1.4. Framework

2.1.4.1. Pengertian Framework

Menurut Gungun Septian (2011: 7) *framework* sebagaimana arti dalam bahasa Indonesia yaitu kerangka kerja yang dapat diartikan sebagai kumpulan dari *library (class)* yang dapat diturunkan, atau dapat langsung dipakai fungsinya oleh modul-modul atau fungsi yang akan kita kembangkan.

Keuntungan menggunakan *framework* adalah:

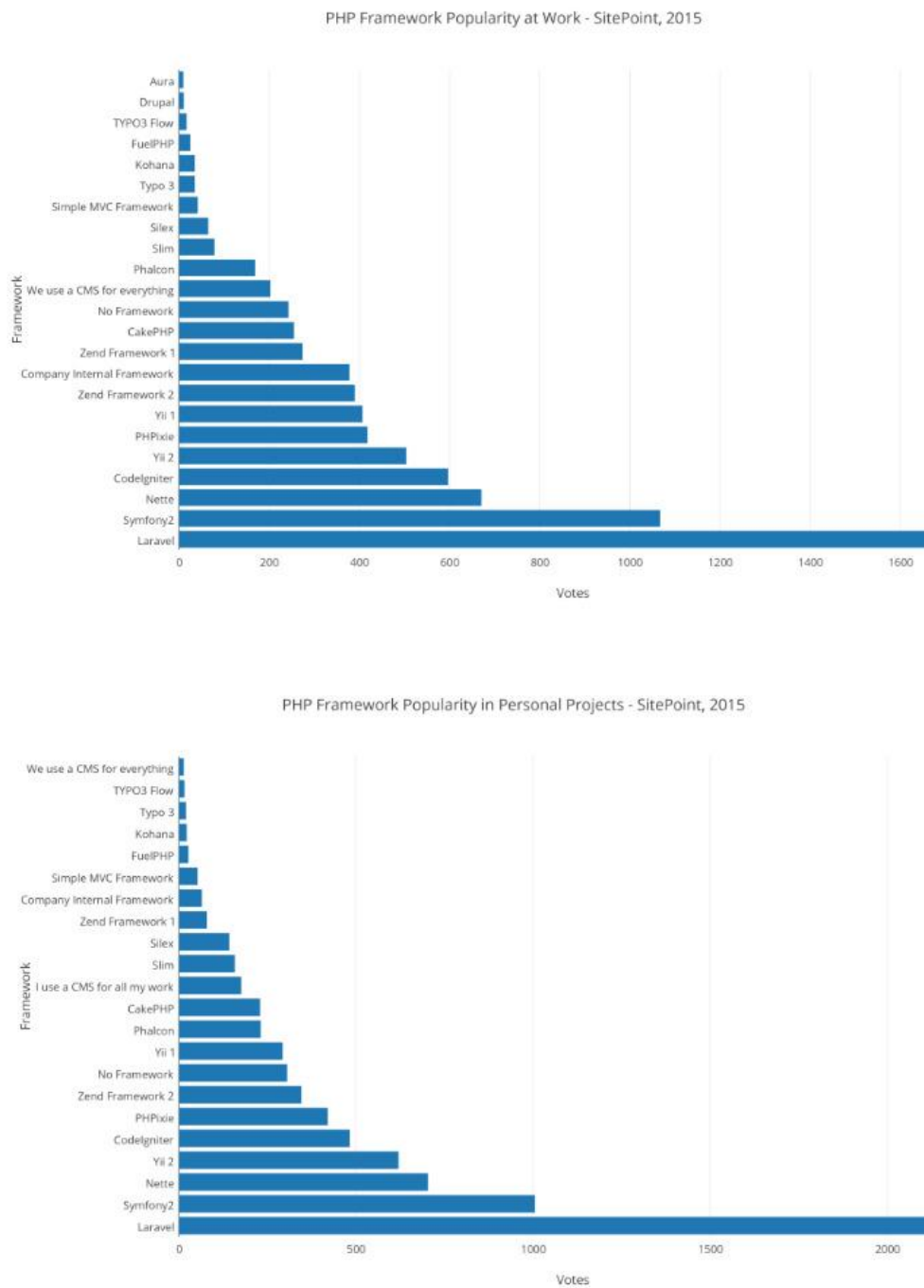
1. Waktu pembuatan *website* jauh lebih singkat. Kode aplikasi *website* menjadi lebih mudah dibaca, karena sedikit dan sifatnya pokok.
2. *Website* jadi lebih mudah diperbaiki, karena tidak perlu fokus kesemua komponen kode *website*, terutama kode *system framework*.
3. Tidak perlu lagi membuat kode penunjang aplikasi *website* seperti koneksi *database*, validasi *form*, GUI dan keamanan.
4. Pikiran menjadi lebih terfokus terhadap kode alur permasalahan *website*, apa yang ditampilkan dan layanan apa saja yang diberikan dari aplikasi tersebut.

5. Jika dikerjakan secara *team work*, maka akan lebih terarah karena *system framework* mengharuskan pengguna menggunakan keteraturan peletakan kode.

2.1.4.2. Laravel

Laravel merupakan *framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. Laravel dibuat oleh *programmer* yang bernama Taylor Otwell dan dirilis dibawah lisensi MIT (*Massachusetts Institute of Technology*) dengan sumber kode yang disediakan di Github. Sama seperti *framework* lainnya, laravel dibangun dengan basis MVC (*Model-View-Controller*). Laravel dilengkapi dengan *command line* tool yang bernama “Artisan” yang bisa digunakan untuk *packaging bundle* dan *instalasi bundle*. Menurut *survey* yang dilakukan oleh sitepoint.com tentang popularitas *framework* PHP di dunia menunjukkan bahwa laravel adalah *framework* yang paling banyak digunakan pada tahun 2014 sampai 2015. *Framework* laravel dirancang sangat baik, sehingga pengguna siapapun dapat dengan menggunakan tanpa harus expert dalam pemrograman. Laravel merupakan *framework* dengan versi PHP yang *uptodate*, karena Laravel mengisyaratkan PHP versi 5.3 ke atas. saat ini Laravel sudah sampai versi 5.2 yang dirilis pada 21 Desember 2015.

Gambar 2.2 merupakan *survey* dari sitepoint.com pada tahun 2014 sampai 2015 mengenai pemakaian *framework*



Gambar 2.2 Survey sitepoin.com

Kelebihan Laravel

1. Menggunakan PHP versi terbaru: dalam PHP versi 5.3 ke atas memiliki cukup banyak fitur baru yang membuat PHP lebih full dan modern.
2. *Expressif*. Laravel adalah sebuah *framework* PHP yang *expressif*, artinya ketika melihat suatu *syntax* Laravel, seorang *programmer* diharapkan akan langsung tahu kegunaan dari *syntax* tersebut meskipun belum pernah mempelajarnya apalagi menggunakannya.
3. *Simple*. Yang membuat Laravel *simple* adalah adanya fitur *Eloquent* ORM. Fitur ini merupakan fitur yang akan menyederhanakan proses *query* ke *database*.
4. *Security*. Aplikasi yang akan dibangun benar-benar aman dari masalah *security* dasar seperti pengamanan *csrf*, autentikasi, sanitasi data, validasi data dan lain-lain. Dan juga bisa didapatkan secara gratis.

Kekurangan Laravel

1. Ukuran data sangat besar dibandingkan dengan php native dikarenakan ada beberapa file yang tidak berguna tidak dihapus misalkan folder *vendor*.

2.1.5. API Doc

API Doc adalah sebuah *tools* untuk mendokumentasikan yang sudah *digenerate* secara otomatis memanfaatkan JSON. API doc sangat berguna untuk penulis dalam mendokumentasikan semua *service* yang ada di *website* penggalangan dana Wake Up Indonesia.

2.1.6. Postman

Postman merupakan sebuah *REST client* berbasis *web* yang tersedia dalam bentuk ekstensi pada google chrome.

Postman juga berguna untuk penulis dalam mengembangkan *REST web service* dan digunakan untuk pengujian. *Postman* juga digunakan untuk menguji sebuah *endpoint web service*.

2.1.7. Wake Up Indonesia

2.1.7.1. Profil Wake Up Indonesia

Wake Up Indonesia Sebuah gerakan pemuda Indonesia yang memiliki visi untuk membangun Indonesia melalui kontribusi nyata yang mereka bisa kerjakan. Gerakan Wake Up Indonesia memiliki program-program kepemudaan yang meliputi berbagai bidang, di antaranya edukasi, lingkungan, olahraga, seni budaya, dan sosial. Gerakan ini bersifat sosial, memberikan penyadaran kepada pemuda agar turut serta dalam membangun negeri ini melalui hal-hal sederhana yang mereka bisa. “Sudah saatnya kita bangun dan bergerak untuk Indonesia”. Kami memulai gerakan ini dengan melalui hal yang kami bisa, yaitu seni. Fokus dari seluruh program kami adalah penggalangan dana untuk membangun pembangkit listrik tenaga angin di pulau-pulau terpencil dan terluar yang membutuhkan di Indonesia.

2.1.7.2. Program Kerja

Program

1. #WIGoestoCampus

Acara talk show keliling yang dilakukan dari kampus ke kampus dengan tujuan memberikan penyadaran dan wawasan terhadap pemuda. #WIGoestoCampus akan mengundang praktisi, seniman, akademisi, negarawan, tokoh nasional, dan tokoh inspiratif lainnya untuk berbagi pengetahuan serta pengalaman dalam membangun Indonesia. Harapannya, setelah mengikuti acara ini para pemuda yang hadir dapat mendapatkan inspirasi dan mulai sadarkan dirinya untuk selanjutnya membangun Indonesia. #WIGoestoCampus juga sebagai ajang kampanye gerakan Wake Up Indonesia yang mengajak lebih banyak lagi pemuda untuk memberika suatu kontribusi bagi negerinya.

2. #WI.com

Website #WI.com diwujudkan dalam upaya menjangkau masyarakat yang lebih luas agar terlibat berpartisipasi dalam gerakan ini. Di dalamnya, selain memuat profil gerakan Wake Up Indonesia, nantinya juga akan diberikan akses bagi para donatur untuk bersedekah secara *online*. Adapun pendaftaran menjadi relawan WI juga akan dibuat *online* melalui *website* ini. Diharapkan melalui *website* ini kami bisa menjangkau lebih banyak massa, baik di Indonesia maupun juga di luar Indonesia.

3. #WINGamen

Kegiatan ini adalah ajang apresiasi kesenian, sekaligus penggalangan dana dan kampanye gerakan Wake Up Indonesia, di tempat-tempat strategis yang sering dikunjungi oleh banyak pemuda. Program ini bertujuan untuk

lebih menyentuh dan mengajak lebih banyak pemuda untuk mendukung dan berpartisipasi dalam gerakan Wake Up Indonesia.

4. #RelawanWI

Sebagai sarana untuk mengakomodasi pemuda-pemuda yang ingin berpartisipasi dalam pembangunan negeri. Para relawan nantinya akan ditempatkan di dalam program sesuai dengan keahlian yang dimilikinya dan turut serta dalam mendukung program Wake Up Indonesia.

5. #WIColourFunRunning

Olahraga lari saat ini sedang sangat diminati banyak orang, karenanya untuk menghimpun dukungan secara massal dan mengampanyekan gerakan tersebut ke berbagai kalangan kami membuat program #WIColourFunRunning. Program ini akan diadakan pada saat *car free day* di Jalan Sudirman dan Bundaran Hotel Indonesia (Jakarta).

6. #SeniUntukNegeri

Gerakan Wake Up Indonesia dimulai dari mahasiswa yang bergerak dalam bidang kesenian karenanya kami memulai dari hal yang paling kami bisa, yaitu melalui seni. #SeniUntukNegeri mengajak seluruh seniman di Indonesia untuk melakukan suatu sumbangsih bagi negerinya. Kami mengajak seniman dari berbagai latar belakang keseniannya, di antaranya pelukis, penyanyi, *group ensemble*, aktor, teater, street art, penari, deklamator, ddesainer, perupa, *orchestra*, manajemen pertunjukan, dll. untuk berpartisipasi dalam membangun Indonesia. Konser dan pameran #SeniUntukNegeri akan kami adakan di Gedung Pertunjukan dan seluruh uang tiket yang terjual akan kami donasikan untuk pembangunan

Pembangkit Listrik Tenaga Angin di pulau-pulau terluar dan terpencil yang membutuhkan di Indonesia.

7. #RisetUntukNegeri

Merupakan kampanye yang ditujukan untuk mengajak para akademisi dalam mengabdikan ilmunya kepada negeri. Akademisi akan membimbing dan memberikan arahan bagi relawan yang ditugaskan untuk membangun pulau dalam program #EkspedisiPemudaNusantara. Riset pendahuluan yang akan meneliti kebutuhan-kebutuhan pulau akan dilakukan oleh para relawan sebelum melaksanakan program. Program-program yang disusun, diterapkan dengan penyesuaian berdasarkan riset pendahuluan yang dilakukan.

8. #EkspedisiPemudaNusantara

Ini adalah program utama dari gerakan Wake Up Indonesia. Khusus program ini, kami akan menempuh waktu selama dua setengah tahun termasuk persiapan dan perekrutan relawan. Program ini dimulai dengan riset pendahuluan dengan memetakan kebutuhan yang paling utama dari pulau yang akan diberikan program sampai dengan pendampingan selama tiga bulan. #EkspedisiPemudaNusantara akan mengirimkan pemuda-pemuda terbaik yang akan diseleksi menjadi sepuluh pemuda yang kami sebut *Team 10* untuk membangun pulau selama dua tahun. Program-program yang akan diberikan disusun berdasarkan kebutuhan dari pulau yang dikunjungi. Seluruh program tersebut memiliki tujuan akhir untuk membentuk kemandirian baik dalam segi ekonomi, lingkungan, sosial, budaya, dan pendidikan. Di akhir program, akan dibentuk Team 10 versi

pulau untuk meneruskan dan menjaga pembangunan di pulau yang diberikan program dan menjadi mitra kerja Wake Up Indonesia.

2.1.7.3. Penggalangan Dana

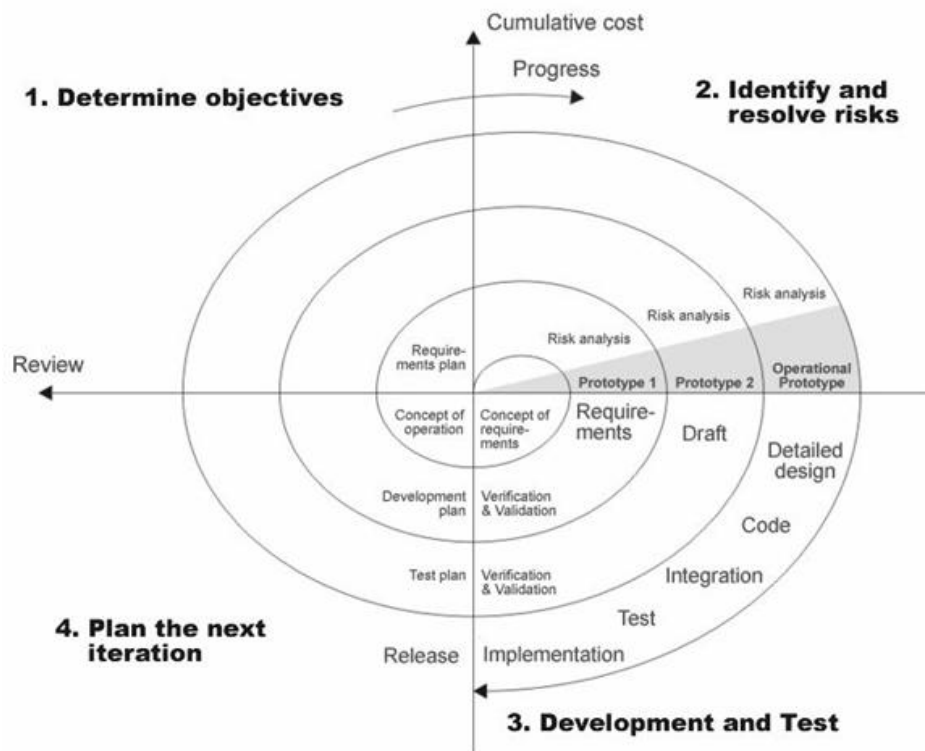
Penggalangan dana dibagi dari 2 suku kata yaitu penggalangan dan dana. Menurut KBBI (Kamus Besar Bahasa Indonesia) Penggalangan /peng-ga-la-ngan/ adalah proses atau cara dan dana /da-na/ adalah uang yang diperlukan untuk suatu keperluan. Penggalangan dana dalam konteks penelitian ini menurut penulis adalah cara mengumpulkan dana dari *donator* yang bersifat sosial dengan *web* sebagai media pengumpulan dana. Penggalangan dana termasuk ke-dalam program kerja Wake Up Indonesia yang ada di *wi.com*.

2.1.8. Pengembangan Web Service Menggunakan Metode Spiral

Salah satu metodologi pengembangan sistem yang digunakan dalam penelitian ini adalah metode Spiral (*Spiral model*). Metode ini ditemukan sekitar tahun 1988 oleh Barry Boehm pada artikel *A Spiral Model of Software Development and Enhancement*. Model Spiral adalah salah satu bentuk evolusi yang menggunakan metode iterasi natural yang dimiliki oleh model *prototyping* dan digabungkan dengan aspek sistematis yang dikembangkan dengan model *Waterfall*. Tahap desain umumnya digunakan pada model *Waterfall*, sedangkan tahap *prototyping* adalah suatu model dimana *software* dibuat *prototype* (*incomplete model*). Jika *prototype*-nya sudah sesuai dengan keinginan *user*, maka proses *Software Engineering* dilanjutkan dengan membuat produk sesungguhnya dengan menambah dan memperbaiki kekurangan dari *prototype* tadi.

Model Spiral dibagi menjadi beberapa *framework* aktivitas, yang disebut dengan *task regions*. Kebanyakan aktivitas-aktivitas tersebut dibagi antara 3 hingga 4 aktivitas. Berikut adalah aktivitas-aktivitas yang dilakukan dalam model Spiral:

1. **Planning.** Aktivitas perencanaan ini dibutuhkan untuk menentukan sumberdaya, perkiraan waktu pengerjaan, dan informasi lainnya yang dibutuhkan untuk pengembangan *software*.
2. **Risk Analysis.** Aktivitas analisis risiko ini dijalankan untuk menganalisis baik risiko secara teknis maupun secara manajerial. Tahap inilah yang mungkin tidak ada pada model proses yang juga menggunakan metode iterasi, tetapi hanya dilakukan pada model Spiral.
3. **Engineering.** Aktivitas yang dibutuhkan untuk membangun 1 atau lebih representasi dari aplikasi secara teknis.
4. **Customer evaluation.** Aktivitas yang dibutuhkan untuk mendapatkan *feedback* dari *user* berdasarkan evaluasi mereka selama representasi *software* pada tahap engineering maupun pada implementasi selama instalasi *software* pada tahap *construction and release*.



Gambar 2.3 Aktivitas Metode Spiral

Dilihat berdasarkan aktivitas-aktivitas yang dilakukan dalam model Spiral, maka dapat disimpulkan bahwa model Spiral memiliki kelebihan sebagai berikut:

1. Penggunaan *prototype* pada setiap siklus mengurangi risiko di dalam evolusi produk.
2. Penanganan langsung terhadap risiko teknis dapat mencegah berkembangnya risiko menjadi permasalahan yang serius.
3. Pengembang dapat lebih mudah memahami dan bereaksi terhadap risiko setiap tingkat evolusi.

Adapun kekurangan dari model Spiral dapat disimpulkan sebagai berikut:

1. Diperlukan adanya tenaga ahli untuk memperkirakan risiko yang akan dihadapi.
2. Diperlukan perkiraan risiko yang masuk akal.

2.2. Kerangka Berpikir

Langkah pertama yang dilakukan dalam penelitian ini adalah melakukan observasi mengenai penggalangan dana oleh yayasan Wake Up Indonesia. Penggalangan dana dilakukan secara manual, meminta sumbangan ke berbagai lembaga atau individu. Dalam meminta sumbangan ini sering terjadi kendala yaitu lembaga-lembaga tidak mempercayai pihak Wake Up Indonesia tentang penggalangan dana tersebut.

Langkah kedua yaitu penulis bertemu dengan ketua yayasan Wake Up Indonesia yang bernama Irfan Hakim membahas tentang pembuatan *website* Wake Up Indonesia. Pembuatan *website* ini memiliki berbagai fitur, fitur-fitur ini yang nantinya akan dikembangkan menggunakan *web service* untuk mengelola sumber daya data yang dibutuhkan oleh *website* Wake Up Indonesia.

Langkah selanjutnya adalah menentukan alur serta mekanisme yang akan digunakan, sehingga ditentukan pengembangan metode spiral. Adapun pengembangan menggunakan metode spiral lebih menitik beratkan pada analisis kebutuhan sistem dan umpan balik dari pengguna sebagai acuan dalam melakukan proses *coding*. Setelah analisis dilakukan, maka dilakukan pengimplementasian kode (*coding*). *Coding* dilakukan menggunakan *framework*. *Framework* laravel untuk *back-end* dengan aplikasi *Integrated Development Environment* (IDE) *Netbeans* versi 8.0.1.

Langkah selanjutnya adalah proses uji coba. Tujuan diadakan pengujian adalah untuk mengetahui apakah *web service* telah berjalan dengan baik sesuai yang diharapkan atau masih terdapat kesalahan pada implementasi sistem. Untuk itu peneliti menggunakan *Unit Testing* untuk pengujian yang hanya memperhatikan hasil keluaran dari sistem secara fungsional dan *bug fixing* guna memperbaiki kesalahan pada *web service*.

Langkah terakhir dipenelitian ini yaitu penyusunan laporan hasil penelitian. Dari kerangka berpikir yang telah dijabarkan, maka penanganan masalah-masalah yang ada dapat diatasi dengan perencanaan *web service* yang baik dengan memperhatikan kebutuhan fungsional secara khusus. Gambaran dari kerangka berpikir dapat dilihat di Gambar 2.4



Gambar 2.4 Kerangka Berpikir Penelitian

BAB III

METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

Penelitian dilakukan di Yayasan Wake Up Indonesia, Sekretariat Wake Up Indonesia yang terdapat di Universitas Negeri Jakarta sejak bulan September 2015 hingga Desember 2015.

3.2. Prosedur Penelitian

Secara garis besar, prosedur penelitian dapat dijelaskan sebagai berikut :

1. Melakukan observasi dan wawancara di sekretariat Yayasan Wake Up Indoensia.
2. Melakukan analisa kebutuhan baik secara fungsional maupun non fungsional.
3. Implementasi *coding* untuk membuat *service* menggunakan aplikasi IDE Netbeans versi 8.0 yang dibuat dengan menggunakan *framework*, yaitu laravel untuk bagian *back-end* disertai pembuatan rancangan *database* menggunakan MySQL PhpMyAdmin 5.6.24.
4. Melakukan uji coba *web service* oleh penguji ahli.
5. Mengisi Tabel kriteria pengujian pada setiap *endpoint web service*.
6. Menyimpulkan hasil dari penelitian.

3.3. Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini adalah metode penelitian dan pengembangan (Research & Development atau R&D). Menurut Sugiyono (2009: 40) metode R&D adalah metode penelitian yang mengacu pada upaya yang diperlukan untuk menciptakan produk baru dan menguji keefektifan produk tersebut. Metode pengembangan yang digunakan adalah Metode Spiral. Metode Spiral adalah model proses pengembangan *software* evolusioner yang menggabungkan pendekatan prototyping yang sifatnya iteratif dengan aspek-aspek sistematis dan terkendali seperti yang dijumpai pada metode waterfall. Penggunaan metode spiral dalam penelitian ini terdiri dari 4 aktivitas, yaitu:

1. Perencanaan (*Planning*).
2. Analisis risiko (*Risk Analysis*), analisis alternatif dan identifikasi/pemecahan risiko.
3. Rekayasa (*Engineering*), pengembangan level berikutnya dari produk.
4. Evaluasi Pemakai (*Customer Evaluation*) penilaian terhadap hasil rekayasa.

3.4. Perencanaan dan Perancangan *Web Service*

3.4.1. Perencanaan (*Planning*).

3.4.1.1. Pengumpulan Data.

Tahapan perencanaan diawali dengan pengambilan atau pengumpulan data tentang bagaimana proses pembuatan *web service* di *website* penggalangan

dana Wake Up Indonesia. Pengumpulan data yang digunakan adalah: observasi, wawancara, dan dokumentasi.

3.4.1.2. Observasi

Observasi merupakan proses untuk memperoleh data dari tangan pertama dengan mengamati orang dan tempat pada saat dilakukan penelitian (Sugiyono, 2014:197). Dengan observasi penulis mengetahui proses pembuatan *website* menggunakan *web service*.

3.4.1.3. Wawancara

Wawancara digunakan sebagai teknik pengumpulan data apabila penulis akan melakukan studi pendahuluan untuk menemukan permasalahan yang harus diteliti. Pengumpulan data melalui wawancara dilakukan dengan mewawancarai pendiri Wake Up Indonesia Irfan Hakim di sekretariat Wake Up Indonesia mengenai kebutuhan fungsional *web service* di *web* Wake Up Indonesia.

3.4.1.4. Rancangan Kebutuhan Fungsional

Tabel 3.1 Tabel *User Non Member*

Daftar Kebutuhan	
1	Dapat Melakukan daftar sebagai <i>user member</i>
2	Tidak dapat melakukan daftar apabila alamat e-mail sudah terpakai
3	Melakukan donasi
4	Melihat donatur di berbagai <i>project</i>
5	Melihat relawan di berbagai <i>project</i>
6	Melihat semua <i>project</i> penggalangan dana Wake Up Indonesia
7	Melakukan cek status donasi untuk terverifikasinya sebuah donasi
8	Menampilkan semua <i>project</i>
9	Menampilkan <i>project</i> berdasarkan kategori
10	Menampilkan sebuah donasi masuk per bulan
11	Menampilkan semua <i>project</i> secara partial
12	Menampilkan sebuah donasi masuk per bulan per kategori

Tabel 3.2 Tabel *User Member*

Daftar Kebutuhan	
1	Dapat mengajukan diri menjadi relawan di berbagai <i>project</i> dengan ketentuan syarat tertentu
2	Melakukan donasi
3	Melihat donatur di berbagai <i>project</i>
4	Melihat relawan di berbagai <i>project</i>
5	Melihat semua <i>project</i> penggalangan dana Wake Up Indonesia
6	Menampilkan dia telah menjadi donator di berbagai <i>project</i>
7	Menampilkan dia telah menjadi relawan di berbagai <i>project</i>
8	Menampilkan halaman profile
9	Mengubah halaman profile
10	Melakukan pergantian password
11	Mengubah foto profile
12	Melakukan cek status donasi untuk terverifikasinya sebuah donasi
13	Menampilkan semua <i>project</i>
14	Menampilkan <i>project</i> berdasarkan kategori
15	Menampilkan semua <i>project</i> secara partial
16	Menampilkan sebuah donasi yang masuk per bulan
17	Melakukan reset password

3.4.1.5. Rancangan *Database*

Nama *database* dalam *website* penggalangan dana Wake Up Indonesia adalah ‘wui’ yang terdiri dari tujuh tabel dan dibuat menggunakan *MySQL*. Dengan *tools phpMyAdmin*. *Database* dikembangkan untuk mendukung kebutuhan sistem dalam menyimpan data *user dan admin* lalu menampilkan kembali data *user dan admin* sebagai informasi bagi pihak Wake Up Indonesia. Berikut di bawah ini akan dijelaskan struktur data dari *database* ‘wui’ beserta perannya dalam *Entity Relationship Diagram (ERD)*. “Entity Relationship Diagram merupakan suatu teknik dokumentasi yang digunakan untuk menyajikan relasi antara entitas (sumber daya, peristiwa dan agen) dalam sebuah sistem” (Hall, James A, 2011:10). Berikut akan dijelaskan struktur dari *database* ‘wui’.

Tabel 3.3 Struktur Data Tabel Admin

No	Nama Atribut	Tipe Atribut	Keterangan
1	id	int (10)	Not Null
2	username	varchar (20)	Not Null
3	nama_depan	varchar (25)	Not Null
4	nama_belakang	varchar (25)	Not Null
5	email	varchar (255)	Primary Key
6	password	text	Not Null
7	foto	text	Not Null
8	job	varchar (15)	Not Null
9	in_charge	varchar (10)	Not Null

Tabel 3.4 Struktur Data Tabel Donasi

No	Nama Atribut	Tipe Atribut	Keterangan
1	id_donasi	int (10)	Primary Key
2	id_project	int (10)	Not Null
3	id_user	int (10)	Not Null
4	email	varchar (50)	Not Null
5	nama_depan	varchar (20)	Not Null
6	nama_belakang	varchar (30)	Not Null
7	telp	varchar (15)	Not Null
8	nominal	int (10)	Not Null
9	ref_code	varchar (10)	Not Null
10	donation_prepared	timestamp	Not Null
11	donation_status	enum('0','1')	Not Null
12	donation_paid	datetime	Not Null

Tabel 3.5 Struktur Data Tabel Kategori

No	Nama Atribut	Tipe Atribut	Keterangan
1	id	int (11)	Primary Key
2	kategori	varchar (25)	Not Null

Tabel 3.6 Struktur Data Tabel Pengguna

No	Nama Atribut	Tipe Atribut	Keterangan
1	id_user	int (10)	Primary Key
2	nama_depan	varchar (20)	Not Null
3	nama_belakang	varchar (30)	Not Null
4	email	varchar (50)	Not Null
5	telp	varchar (15)	Not Null
6	jenis_kelamin	enum ('L', 'P')	Not Null

7	foto	text	Not Null
8	about	text	Not Null
9	remember_token	text	Not Null
10	created_at	timestamp	Not Null

Tabel 3.7 Struktur Data Tabel *Project*

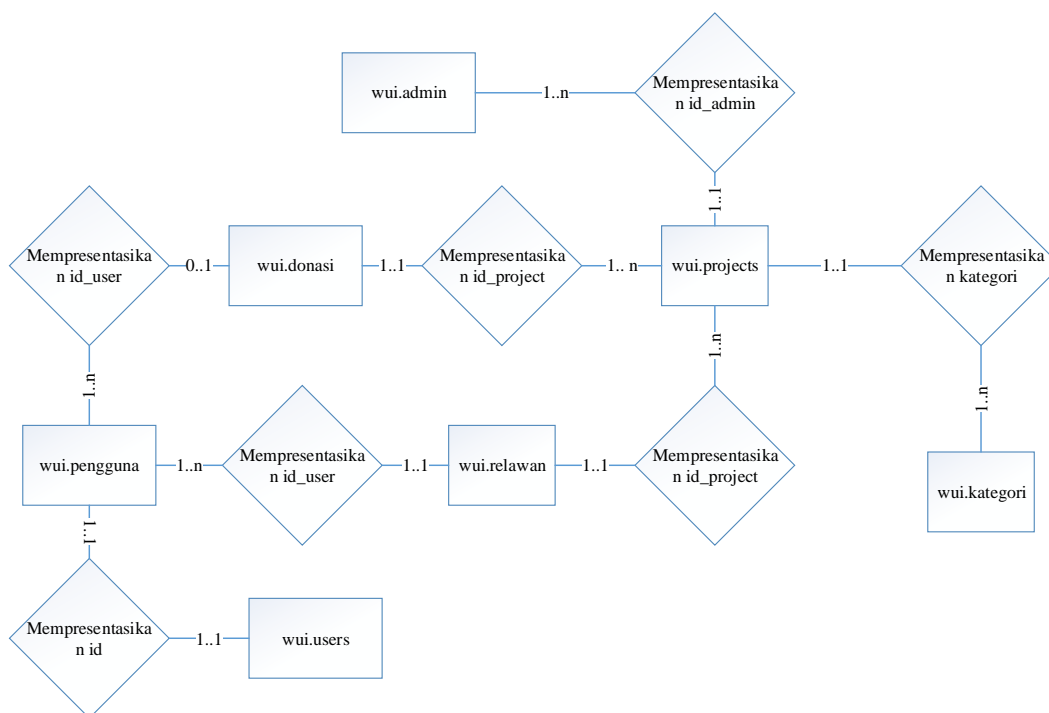
No	Nama Atribut	Tipe Atribut	Keterangan
1	id_project	int (10)	Primary Key
2	nama_project	varchar (50)	Not Null
3	deskripsi_project	text	Not Null
4	kategori	varchar (2)	Not Null
5	id_admin	varchar (10)	Not Null
6	project_created	timestamp	Not Null
7	target_nominal	int (10)	Not Null
8	project_deadline	date	Not Null
9	foto	text	Not Null

Tabel 3.8 Struktur Data Tabel *Relawan*

No	Nama Atribut	Tipe Atribut	Keterangan
1	id_relawan	varchar (10)	Primary Key
2	id_project	varchar (10)	Not Null
3	id_user	varchar (10)	Not Null
4	status	enum ('0','1')	Not Null

Tabel 3.9 Struktur Data Tabel *Users*

No	Nama Atribut	Tipe Atribut	Keterangan
1	id	int (10)	Primary Key
2	email	varchar (255)	Not Null
3	password	text	Not Null
4	verified	enum ('0','1')	Not Null



Gambar 3.1 ERD Database Website Penggalangan Dana Wake Up Indonesia

3.4.2. Analisis Kebutuhan dan Risiko

Kebutuhan fungsional harus mendefinisikan aksi dasar yang diambil oleh aplikasi untuk menerima dan memproses input sehingga menghasilkan *output*, sedangkan analisis risiko mendefinisikan kemungkinan kesalahan yang terjadi pada *web service*. Pada penelitian ini, terdapat kebutuhan fungsional utama yang dapat dilihat pada Tabel 3.10

Tabel 3.10 Daftar Kebutuhan Fungsional Sistem dan Analisis Risiko

No	Kebutuhan Fungsional	Analisis Risiko	Actor
1	<i>Login</i>	a. <i>User</i> gagal login b. <i>User</i> lupa password	<i>User Member</i>
2	Cek status Donasi	a. <i>User</i> lupa kode referensi b. <i>User</i> lupa email	<i>All User</i>
3	Mengajukan Donasi Sebuah	a. <i>Project</i> telah terdanai b. <i>Project</i> melewati tanggal deadline	<i>All User</i>

Project			
4	Mengajukan Relawan Dalam Project	a. Syarat sebagai relawan tidak layak	<i>User Member</i>
5	Ganti Password	a. <i>User</i> gagal mengganti password	<i>User Member</i>
6	Daftar	a. <i>User</i> gagal melakukan daftar member	<i>All User</i>
7	Kode Verifikasi Akun	a. <i>User</i> tidak menerima kode verifikasi b. <i>User</i> salah memasukan email	<i>All User</i>

3.4.2.1. Kriteria Keberhasilan Sistem

Dalam melakukan pengolahan data pada *web service* yang dibuat, dibutuhkan beberapa tahap agar *output* dari *endpoint* sesuai dengan fungsinya, oleh karena itu, dibuatlah kriteria untuk *web service*. *Web service* dikatakan berhasil jika memenuhi semua kriteria tersebut seperti ditunjukkan pada Tabel 3.13.

Tabel 3.13 Kriteria Keberhasilan Web Service

No	Kriteria Keberhasilan
1	<i>Web service</i> dapat mengotentikasi <i>user frontend web</i> Wake Up Indonesia
2	<i>Web service</i> dapat memetakan sejumlah <i>requirement</i> ke dalam sebuah <i>endpoint</i> .
3	<i>Web service</i> menentukan tipe <i>request</i> yang harus digunakan <i>frontend</i> sesuai kebutuhan.
4	<i>Web service</i> dapat memberi <i>respon</i> data sesuai dengan permintaan dari <i>frontend</i> .
5	<i>Web service</i> dapat digunakan oleh <i>cross platform frontend</i>

3.5. Tipe Request

Tipe *request* yang digunakan dalam penelitian ini dibagi menjadi 3 tipe *request* yaitu *get*, *post*, dan *put*.

3.5.1. GET

Tabel 3.14 Kebutuhan Fungsional Tipe Request Get

No	Kebutuhan fungsional
1	Menampilkan seluruh <i>project</i> secara partial
2	Mengirim kode verifikasi akun
3	Menampilkan profil <i>user</i>
4	Menampilkan record donasi <i>user</i>
5	Melakukan pencarian <i>project</i> berdasarkan <i>keyword</i>
6	Menampilkan <i>project</i> berdasarkan kategori secara partial
7	Menampilkan data sebuah <i>project</i>
8	Melakukan pengecekan apakah masih dapat melakukan donasi pada

sebuah *project*

3.5.2. POST

Tabel 3.15 Kebutuhan Fungsional Tipe Request Post

No	Kebutuhan fungsional
1	Mengajukan donasi untuk sebuah <i>project</i>
2	Mengajukan donasi untuk sebuah <i>project</i> (<i>Non Member</i>)
3	Melakukan cek terhadap status donasi
4	Mengajukan diri sebagai relawan dalam <i>project</i>
5	Otentikasi <i>user</i>
6	Mendaftarkan <i>user</i>
7	Mengirim ulang email verifikasi akun <i>user</i>
8	Melakukan reset password

3.5.3. PUT

Tabel 3.16 Kebutuhan Fungsional Tipe Request Put

No	Kebutuhan fungsional
1	<i>Update</i> nama depan dan nama belakang <i>user</i>
2	Ganti password <i>user</i>
3	Upload foto <i>user</i>

3.6. Hasil Tipe Request

3.6.1. Method Get

Tabel 3.17 Hasil Endpoint Method Get

No	Kebutuhan fungsional	Endpoint
1	Menampilkan seluruh <i>project</i> secara partial	project/{offset}/{limit}
2	Mengirim kode verifikasi akun	verifyAcc?q={key}
3	Menampilkan profil <i>user</i>	profile/{id_user}
4	Menampilkan record donasi <i>user</i>	showDonation/{id_user}
5	Melakukan pencarian <i>project</i> berdasarkan <i>keyword</i>	project/cari?key={key}
6	Menampilkan <i>project</i> berdasarkan kategori secara partial	project/{kategori}/{offset}/{limit}
7	Menampilkan data sebuah <i>project</i>	project/show/{id_project}
8	Melakukan pengecekan donasi	project/isAllowed/{id_project}

apakah masih dapat melakukan donasi pada sebuah *project*

3.6.2. Method *Post*

Tabel 3.18 Hasil *Endpoint* Method *Post*

No	Kebutuhan fungsional	<i>Endpoint</i>
1	Mengajukan donasi untuk sebuah <i>project</i>	project/donate
2	Mengajukan donasi untuk sebuah <i>project (Non Member)</i>	project/donateX
3	Melakukan cek terhadap status donasi	project/statusDonasi
4	Mengajukan diri sebagai relawan dalam <i>project</i>	project/daftarRelawan
5	otentikasi <i>user</i>	authenticate
6	Mendaftarkan <i>user</i>	daftar
7	Mengirim ulang email verifikasi akun <i>user</i>	resendEmail
8	Melakukan reset password	resetPassword

3.6.3. Method *Put*

Tabel 3.19 Hasil *Endpoint* Method *Get*

No	Kebutuhan fungsional	<i>Endpoint</i>
1	<i>Update</i> nama depan dan nama belakang <i>user</i>	update/{id_user}
2	Ganti password <i>user</i>	editPass
3	Upload foto <i>user</i>	uploadGambarUser

3.7. Rekayasa Pengembangan

3.7.1. Rekayasa Pengembangan Sistem

Dalam mengembangkan *web service* membutuhkan dukungan perangkat keras (*hardware*) dan *software* (*software*). Untuk *backend* menggunakan *framework* laravel dan MySQL untuk *database*. Spesifikasi perangkat keras dan *software* yang digunakan untuk pengembangan *web service* adalah :

1. Perangkat Keras

Dalam penelitian ini digunakan sebuah laptop dengan spesifikasi sebagai berikut:

- a. Windows 7 Ultimate SP1 32bit
- b. Intel(R) Pentium(R) CPU P6100 @ 2.00Ghz (2 CPUs), 2.0Ghz
- c. 3072MB RAM
- d. Intel(R) HD Graphics 256MB

2. *Software*

Software yang digunakan dalam penelitian ini antara lain:

- a. NetBeans IDE 8.0.1
- b. JDK 1.8.0
- c. XAMPP 5.6.8
- d. Apache 2.4.12
- e. PHP 5.6.8
- f. PhpMyAdmin 4.3.11

3.7.2. Evaluasi

Pengujian *web service* menggunakan metode *unit testing* yaitu pengujian yang digunakan terhadap nilai keluaran (*output*) dari *input* url. Metode ini digunakan untuk mengetahui apakah *web service* berfungsi berdasarkan kriteria keberhasilan program secara fungsional sehingga kesalahan dari *web service* dalam memenuhi kebutuhan dapat diketahui. Pengujian kebutuhan

fungsional dilakukan dengan proses *scenario* yang telah ditentukan seperti tabel 3.20 – tabel 3.38 menggunakan ekstensi *postman* di google chrome.

Tabel 3.20 Pengujian Fungsional pada *Endpoint* Dengan tipe *Request Get* Menampilkan Seluruh *Project* Secara Partial

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	Project/{offset}/ {limit}		a. Bila <i>endpoint</i> benar maka akan menampilkan data <i>id_project</i> , <i>nama_project</i> , <i>deskripsi_project</i> , <i>kategori</i> , <i>id_admin</i> , <i>project_created</i> , <i>target_nominal</i> , <i>project_deadline</i> , foto dan status <i>respon</i> 202 OK b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

Tabel 3.21 Pengujian Fungsional pada *Endpoint* Dengan tipe *Request Get* Mengirim Kode Verifikasi Akun

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	verifyAcc?q={key}		a. Bila <i>endpoint</i> benar maka akan menampilkan result true dan status <i>respon</i> 200 OK b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

Tabel 3.22 Pengujian Fungsional pada *Endpoint* Dengan tipe *Request Get* Menampilkan Profil User

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	profile/{id_user}		a. Bila <i>endpoint</i> benar maka

-
- akan menampilkan data id_user, nama_depan, nama_belakang, email, telp, jenis_kelamin, foto, about, remember_token, created_at, contribution, volunteered dan status respon 200 OK
- b. Bila *endpoint* salah maka status respon 404 Not Found
-

Tabel 3.23 Pengujian Fungsional pada *Endpoint* Dengan tipe *Request Get* Menampilkan Record Donasi User

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	showDonation/{id_user}		<p>a. Bila <i>endpoint</i> benar maka akan menampilkan data id_donasi, id_project, id_user, email, nama_depan, nama_belakang, telp, nominal, ref_code, donation_prepared, donation_status, donation_paid, nama_project status dan status respon 200 OK</p> <p>b. Bila <i>endpoint</i> salah maka status respon 404 Not Found</p>

Tabel 3.24 Pengujian Fungsional pada *Endpoint* Dengan tip *Request Get* Melakukan Pencarian *Project* Berdasarkan *Keyword*

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	project/cari?key={key}		<p>a. Bila <i>endpoint</i> benar maka akan menampilkan data id_project, nama_project, deskripsi_project, kategori, id_admin, project_created, target_nominal, project_deadline, foto dan</p>

-
- status *respon* 202 Accepted
 b. Bila *endpoint* salah maka status *respon* 404 Not Found
-

Tabel 3.25 Pengujian Fungsional pada *Endpoint* Dengan tipe *Request Get* Menampilkan *Project* Berdasarkan Kategori Secara Partial

No	Skenario Proses	Sistem Bekerja (Ya/Tidak) *	Keterangan
1.	project/{kategori}/{offset}/{limit}		a. Bila <i>endpoint</i> benar maka akan menampilkan data <i>id_project</i> , <i>nama_project</i> , <i>deskripsi_project</i> , <i>kategori</i> , <i>id_admin</i> , <i>project_created</i> , <i>target_nominal</i> , <i>project_deadline</i> , <i>foto</i> dan status <i>respon</i> 202 Accepted b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

Tabel 3.26 Pengujian Fungsional pada *Endpoint* Dengan Tipe *Request Get* Menampilkan Data Sebuah *Project*

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	project/show/{id_project}		a. Bila <i>endpoint</i> benar maka akan menampilkan data <i>id_project</i> , <i>nama_project</i> , <i>deskripsi_project</i> , <i>kategori</i> , <i>id_admin</i> , <i>project_created</i> , <i>target_nominal</i> , <i>project_deadline</i> , <i>foto</i> , <i>total_donasi</i> , <i>time_remain</i> , <i>donatur</i> , <i>relawan</i> dan status <i>respon</i> 200 OK b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

Tabel 3.27 Pengujian Fungsional pada *Endpoint* Dengan Tipe *Request Get* Melakukan Pengecekan Apakah Masih Dapat Melakukan Donasi Pada Sebuah *Project*

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	project/isAllowed/{id_project}		a. Bila <i>endpoint</i> benar maka akan menampilkan status true, data dan status <i>respon</i> 200 OK b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

Tabel 3.28 Pengujian Fungsional pada *Endpoint* Dengan Tipe *Request Post* Mengajukan Donasi Untuk Sebuah *Project*

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	project/donate		a. Bila <i>endpoint</i> benar maka akan menampilkan data result, ref_code dan status <i>respon</i> 201 Created b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

Tabel 3.29 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Mengajukan Donasi Untuk Sebuah *Project (Non Member)*

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	project/donateX		a. Bila <i>endpoint</i> benar maka akan menampilkan data result, ref_code dan status <i>respon</i> 201 Created b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

Tabel 3.30 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Melakukan Cek Terhadap Status Donasi

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	project/statusDonasi		a. Bila <i>endpoint</i> benar maka akan menampilkan data,

	email, nama_depan, nama_belakang, telp, nominal, donation_status dan status respon 200 OK
	b. Bila <i>endpoint</i> salah maka status respon 404 Not Found

Tabel 3.31 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Mengajukan Diri Sebagai Relawan Dalam *Project*

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	project/daftarRelawan		a. Bila <i>endpoint</i> benar maka akan menampilkan status true, data dan status respon 201 Created b. Bila <i>endpoint</i> salah maka status respon 404 Not Found

Tabel 3.32 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Otentikasi User

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	authenticate		a. Bila <i>endpoint</i> benar maka akan menampilkan data id_user, nama_belakang, email, telp, jenis_kelamin, foto, about dan status respon 200 OK b. Bila <i>endpoint</i> salah maka status respon 404 Not Found

Tabel 3.33 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Mendaftarkan User

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	daftar		a. Bila <i>endpoint</i> benar maka akan menampilkan status true

- dan status *respon* 201 Created
 b. Bila *endpoint* salah maka status *respon* 404 Not Found

Tabel 3.34 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Mengirim Ulang Email Verifikasi Akun User

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	resendEmail		a. Bila <i>endpoint</i> benar maka akan menampilkan status true dan status <i>respon</i> 200 OK b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

Tabel 3.35 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Melakukan Reset Password

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	resetPassword		a. Bila <i>endpoint</i> benar maka akan menampilkan status true, data dan status <i>respon</i> 200 OK b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

Tabel 3.36 Pengujian Fungsional pada *Endpoint* Dengan *Request Put* Update Nama Depan dan Nama Belakang User

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	update		a. Bila <i>endpoint</i> benar maka akan menampilkan data <i>id_user</i> , <i>nama_depan</i> , <i>nama_belakang</i> , <i>telepon</i> , <i>jenis_kelamin</i> , <i>about</i> dan status <i>respon</i> 200 OK b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

Tabel 3.37 Pengujian Fungsional pada *Endpoint Dengan Request Put Ganti Password User*

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	editPass		a. Bila <i>endpoint</i> benar maka akan menampilkan status result true dan status <i>respon</i> 200 OK b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

Tabel 3.38 Pengujian Fungsional pada *Endpoint Dengan Request Put Upload Foto User*

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	uploadGambarUser		a. Bila <i>endpoint</i> benar maka akan menampilkan data foto dan status <i>respon</i> 200 OK b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Hasil Penelitian

Pengembangan menghasilkan 19 *endpoint*. Setiap *endpoint* dikembangkan satu per satu mulai dari penentuan algoritma, penulisan kode program dan pengujian. setelah sebuah *endpoint* berhasil dikembangkan dengan baik tanpa kesalahan lalu dilanjutkan membuat *endpoint* yang baru hingga semua *endpoint* dibuat sesuai dengan fungsinya tanpa ada kesalahan.

4.1.1. Hasil Pengujian Fungsional

Pengujian dengan metode *unit testing* dilakukan dengan menguji *endpoint service* untuk *method* yang digunakan yaitu *method get*, *method post* dan *method put*. Hasil pengujian dapat dilihat pada Tabel 4.1 sampai dengan Tabel 4.19.

4.1.1.1. Hasil Pengujian Fungsional Method Get

Tabel 4.1 Pengujian Fungsional pada *Endpoint* Dengan tipe *Request Get* Menampilkan Seluruh *Project* Secara Partial

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	project/0/3	Ya	<ol style="list-style-type: none">Bila <i>endpoint</i> benar maka akan menampilkan data <i>id_project</i>, <i>nama_project</i>, <i>deskripsi_project</i>, <i>kategori</i>, <i>id_admin</i>, <i>project_created</i>, <i>target_nominal</i>, <i>project_deadline</i>, <i>foto</i> dan status <i>respon</i> 202 OKBila <i>endpoint</i> salah maka status <i>respon</i> 404 Not FoundBerikut isi dari JSON:

```
{
  "result": true,
  "data": [
    {
      "id_project": 1,
      "nama_project": "sedekah
mmr",
      "deskripsi_project":
"project ini bertujuan untuk
menyedahkan sedikit mmr
bagi para fakir mmr",
      "kategori": "2",
      "id_admin": "1",
      "project_created": "2016-
01-10 16:38:21",
      "target_nominal": 4000,
      "project_deadline": "2016-
12-31",
      "foto":
"http://localhost/wui-
service/storage/app/img/projec
ts/project0.jpg"
    },
    {
      "id_project": 2,
      "nama_project":
"#SolikhaHarusSekolah - Dek
Solikhah",
      "deskripsi_project": "The
titles of Washed Out's
breakthrough song and the
first single from Paracosm
share the\n
two most important words in
Ernest Greene's musical
language: feel it. It's a simple
request, as well...",
      "kategori": "3",
      "id_admin": "1",
      "project_created": "2016-
01-10 16:50:02",
      "target_nominal":
2000000,
      "project_deadline": "2016-
12-30",
      "foto":
"http://localhost/wui-
service/storage/app/img/projec
```

```

ts/project1.jpg"
  },
  {
    "id_project": 3,
    "nama_project": "Kopi
Darat Nasional Orang Baik se-
Indonesia",
    "deskripsi_project": "The
titles of Washed Out's
breakthrough song and the
first single from Paracosm
share the two
most important words in
Ernest Greene's musical
language: feel it. It's a simple
request, as well...",
    "kategori": "2",
    "id_admin": "5",
    "project_created": "2016-
01-10 16:38:16",
    "target_nominal":
5000000,
    "project_deadline": "2015-
12-30",
    "foto":
"http://localhost/wui-
service/storage/app/img/projec
ts/project2.jpg"
  }
],
"count": 8
}

```

Tabel 4.2 Pengujian Fungsional pada *Endpoint* Dengan tipe *Request Get* Mengirim Kode Verifikasi Akun

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	verifyAcc?q={key}	Ya	a. Bila <i>endpoint</i> benar maka akan menampilkan result

	<p>true dan status <i>respon</i> 200 OK</p> <p>b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found</p> <p>c. Di Hash dikirim ke <i>frontend</i></p>
--	---

Tabel 4.3 Pengujian Fungsional pada *Endpoint* Dengan tip *Request Get* Menampilkan Profil User

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	profile/1	Ya	<p>a. Bila <i>endpoint</i> benar maka akan menampilkan data <i>id_user</i>, <i>nama_depan</i>, <i>nama_belakang</i>, <i>email</i>, <i>telp</i>, <i>jenis_kelamin</i>, <i>foto</i>, <i>about</i>, <i>remember_token</i>, <i>created_at</i>, <i>contribution</i>, <i>volunteered</i> dan status <i>respon</i> 200 OK</p> <p>b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found</p> <p>c. Berikut isi dari JSON:</p> <pre>{ "result": true, "data": { "id_user": "1", "nama_depan": "Nurmansyah", "nama_belakang": "Santosa", "email": "nurboy@gmail.com", "telp": "0856825957", "jenis_kelamin": "L", "foto": "http://localhost/wui- service/storage/app/img/user s/user1.jpg", "about": "Saya adalah mahasiswa ptik nr 2011", "remember_token":</pre>

```
    null,
    "created_at": "2015-11-07 17:55:26",
    "contribution": [
      {
        "id_project": 1,
        "nama_project":
"sedekah mmr",

"deskripsi_project": "project
ini bertujuan untuk
menyedahkan sedikit mmr
bagi para fakir mmr",
        "foto":
"http://localhost/wui-
service/storage/app/img/proj
ects/project0.jpg"
      },
      {
        "id_project": 6,
        "nama_project":
"Seribu Uang Seratus Ribuan
untuk Daus",

"deskripsi_project": "Project
ini bertujuan membantu
Daus mendapatkan uang
untuk operasi plastik dan
suntik putih agar tidak
jomblo terus",
        "foto":
"http://localhost/wui-
service/storage/app/img/proj
ects/project5.jpg"
      }
    ],
    "volunteered": [
      {
        "id_project": 1,
        "nama_project":
"sedekah mmr",

"deskripsi_project": "project
ini bertujuan untuk
menyedahkan sedikit mmr
bagi para fakir mmr",
        "foto":
"http://localhost/wui-
```

```

service/storage/app/img/projects/project0.jpg"
    },
    {
        "id_project": 6,
        "nama_project":
        "Seribu Uang Seratus Ribuan
        untuk Daus",

        "deskripsi_project": "Project
        ini bertujuan membantu
        Daus mendapatkan uang
        untuk operasi plastik dan
        suntik putih agar tidak
        jomblo terus",
        "foto":
        "http://localhost/wui-
        service/storage/app/img/projects/project5.jpg"
    }
    ]
}
}
}

```

Tabel 4.4 Pengujian Fungsional pada *Endpoint* Dengan tipe *Request* *Get* Menampilkan Record Donasi User

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	showDonation/1	Ya	<p>a. Bila <i>endpoint</i> benar maka akan menampilkan data id_donasi, id_project, id_user, email, nama_depan, nama_belakang, telp, nominal, ref_code, donation_prepared, donation_status, donation_paid, nama_project status dan status <i>respon</i> 200 OK</p> <p>b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found</p> <p>c. Berikut isi dari JSON:</p> <pre> { "result": true, "data": [</pre>

```
{
  "id_donasi": 3,
  "id_project": 1,
  "id_user": 1,
  "email":
  "nurboy@gmail.com",
  "nama_depan":
  "Nurmansyah",
  "nama_belakang":
  "Santosa",
  "telp": "",
  "nominal": 20000,
  "ref_code": "965112",
  "donation_prepared":
  "2016-01-12 13:37:51",
  "donation_status":
  "1",
  "donation_paid":
  "2016-01-12 00:00:00",
  "nama_project":
  "sedekah mmr"
},
{
  "id_donasi": 4,
  "id_project": 6,
  "id_user": 1,
  "email":
  "nurboy@gmail.com",
  "nama_depan":
  "Nurmansyah",
  "nama_belakang":
  "Santosa",
  "telp": "",
  "nominal": 500000,
  "ref_code": "702812",
  "donation_prepared":
  "2016-01-12 14:27:22",
  "donation_status":
  "1",
  "donation_paid":
  "2016-01-13 00:00:00",
  "nama_project":
  "Seribu Uang Seratus Ribuan
  untuk Daus"
},
{
  "id_donasi": 6,
  "id_project": 6,
```

```
"id_user": 1,
"email":
"nurboy@gmail.com",
"nama_depan":
"Nurmansyah",
"nama_belakang":
"Santosa",
"telp": "",
"nominal": 100000,
"ref_code": "484739",
"donation_prepared":
"2016-01-12 15:22:46",
"donation_status":
"0",
"donation_paid":
"0000-00-00 00:00:00",
"nama_project":
"Seribu Uang Seratus Ribuan
untuk Daus"
},
{
"id_donasi": 7,
"id_project": 6,
"id_user": 1,
"email":
"nurboy@gmail.com",
"nama_depan":
"Nurmansyah",
"nama_belakang":
"Santosa",
"telp": "",
"nominal": 100000,
"ref_code": "407986",
"donation_prepared":
"2016-01-12 15:22:46",
"donation_status":
"0",
"donation_paid":
"0000-00-00 00:00:00",
"nama_project":
"Seribu Uang Seratus Ribuan
untuk Daus"
},
{
"id_donasi": 13,
"id_project": 6,
"id_user": 1,
"email":
```

```

"nurboy@gmail.com",
  "nama_depan":
"Nurmansyah",
  "nama_belakang":
"Santosa",
  "telp": "0856825957",
  "nominal": 100000,
  "ref_code": "361916",
  "donation_prepared":
"2016-01-19 15:13:44",
  "donation_status":
"0",
  "donation_paid":
"0000-00-00 00:00:00",
  "nama_project":
"Seribu Uang Seratus Ribuan
untuk Daus"
}
]
}

```

Tabel 4.5 Pengujian Fungsional pada *Endpoint* Dengan tip *Request Get* Melakukan Pencarian *Project* Berdasarkan *Keyword*

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	project/cari?key=da us	Ya	<p>a. Bila <i>endpoint</i> benar maka akan menampilkan data <i>id_project</i>, <i>nama_project</i>, <i>deskripsi_project</i>, <i>kategori</i>, <i>id_admin</i>, <i>project_created</i>, <i>target_nominal</i>, <i>project_deadline</i>, foto dan status <i>respon</i> 202 Accepted</p> <p>b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found</p> <p>c. Berikut isi dari JSON:</p> <pre> { "result": true, "data": [{ "id_project": 5, "nama_project": "Daus Nggak Jomblo Lagi", </pre>

```
        "deskripsi_project":
        "Penggalangan dana untuk
        membantu Daus membeli
        mobil agar banyak wanita
        yang mau menjadi
        kekasihnya.",
        "kategori": "1",
        "id_admin": "8",
        "project_created":
        "2016-01-10 16:38:12",
        "target_nominal":
        2000000,
        "project_deadline":
        "2016-12-30",
        "foto":
        "http://localhost/wui-
        service/storage/app/img/projec
        ts/project4.jpg"
    },
    {
        "id_project": 6,
        "nama_project":
        "Seribu Uang Seratus Ribuan
        untuk Daus",
        "deskripsi_project":
        "Project ini bertujuan
        membantu Daus mendapatkan
        uang untuk operasi plastik dan
        suntik putih agar tidak jomblo
        terus",
        "kategori": "2",
        "id_admin": "7",
        "project_created":
        "2016-01-10 16:38:09",
        "target_nominal":
        1000000000,
        "project_deadline":
        "2016-12-31",
        "foto":
        "http://localhost/wui-
        service/storage/app/img/projec
        ts/project5.jpg"
    }
  ]
}
```

Tabel 4.6 Pengujian Fungsional pada *Endpoint* Dengan tipe *Request Get* Menampilkan *Project* Berdasarkan Kategori Secara Partial

No	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	project/1/0/3	Ya	<p>a. Bila <i>endpoint</i> benar maka akan menampilkan data <i>id_project</i>, <i>nama_project</i>, <i>deskripsi_project</i>, <i>kategori</i>, <i>id_admin</i>, <i>project_created</i>, <i>target_nominal</i>, <i>project_deadline</i>, <i>foto</i> dan status <i>respon</i> 202 Accepted</p> <p>b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found</p> <p>c. Berikut isi dari JSON:</p> <pre> { "result": true, "data": [{ "id_project": 4, "nama_project": "Pelajar Indonesia Melawan Asap", "deskripsi_project": "The titles of Washed Out's breakthrough song and the first single from Paracosm share the two most important words in Ernest Greene's musical language: feel it. It's a <i>simple request</i>, as well...", "kategori": "1", "id_admin": "9", "project_created": "2016-01-10 16:38:14", "target_nominal": 50000000, "project_deadline": "2015-12-30", "foto": "http://localhost/wui- service/storage/app/img/project s/project3.jpg" }, { "id_project": 5, </pre>

```

"nama_project": "Daus
Nggak Jomblo Lagi",
"deskripsi_project":
"Penggalangan dana untuk
membantu Daus membeli
mobil agar banyak wanita yang
mau menjadi kekasihnya.",
"kategori": "1",
"id_admin": "8",
"project_created":
"2016-01-10 16:38:12",
"target_nominal":
2000000,
"project_deadline":
"2016-12-30",
"foto":
"http://localhost/wui-
service/storage/app/img/project
s/project4.jpg"
},
{
"id_project": 7,
"nama_project": "test",
"deskripsi_project":
"test aja",
"kategori": "1",
"id_admin": "7",
"project_created":
"2016-01-10 16:38:07",
"target_nominal": 1500,
"project_deadline":
"2015-12-24",
"foto":
"http://localhost/wui-
service/storage/app/img/project
s/project6.jpg"
}
],
"count": 3
}

```

Tabel 4.7 Pengujian Fungsional pada *Endpoint* Dengan Tipe *Request Get* Menampilkan Data Sebuah *Project*

N o	Skenario Proses	Sistem Bekerja (Ya/Tidak)*	Keterangan
1.	project/show/	Ya	a. Bila <i>endpoint</i> benar maka akan

6

-
- menampilkan data `id_project`, `nama_project`, `deskripsi_project`, `kategori`, `id_admin`, `project_created`, `target_nominal`, `project_deadline`, `foto`, `total_donasi`, `time_remain`, `donatur`, `relawan` dan `status`
respon 200 OK
- b. Bila *endpoint* salah maka status *respon* 404 Not Found
- c. Berikut isi dari JSON:
- ```
{
 "result": true,
 "data": {
 "id_project": 6,
 "nama_project": "Seribu
 Uang Seratus Ribuan untuk
 Daus",
 "deskripsi_project":
 "Project ini bertujuan
 membantu Daus mendapatkan
 uang untuk operasi plastik dan
 suntik putih agar tidak jomblo
 terus",
 "kategori": "sosial",
 "id_admin": "7",
 "project_created": "2016-
 01-10 16:38:09",
 "target_nominal":
 1000000000,
 "project_deadline": "2016-
 12-31",
 "foto":
 "http://localhost/wui-
 service/storage/app/img/project
 s/project5.jpg",
 "total_donasi": 600000,
 "time_remain": "346",
 "donatur": [
 {
 "id_user": 1,
 "nama_depan":
 "Nurmansyah",
 "nama_belakang":
 "Santosa",
 "nominal": 500000,
 "ref_code": "702812",
```
-

---

```

 "donation_paid":
"2016-01-13 00:00:00",
 "foto":
"http://localhost/wui-
service/storage/app/img/users/u
ser1.jpg"
 },
 {
 "id_user": 16,
 "nama_depan":
"irfan",
 "nama_belakang":
"hakin",
 "nominal": 100000,
 "ref_code": "994472",
 "donation_paid":
"2016-01-04 00:00:00",
 "foto":
"http://localhost/wui-
service/storage/app/img/users/u
ser16.jpg"
 }
],
 "relawan": [
 {
 "nama_depan":
"irfan",
 "nama_belakang":
"hakin",
 "id_user": 16,
 "foto":
"http://localhost/wui-
service/storage/app/img/users/u
ser16.jpg"
 }
]
}
}

```

---

**Tabel 4.8 Pengujian Fungsional pada *Endpoint* Dengan Tipe *Request Get* Melakukan Pengecekan Apakah Masih Dapat Melakukan Donasi Pada Sebuah Project**

| No | Skenario Proses | Sistem Bekerja (Ya/Tidak)* | Keterangan |
|----|-----------------|----------------------------|------------|
|----|-----------------|----------------------------|------------|

|    |                     |    |                                                                                                                                                                                                                                                                     |
|----|---------------------|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | project/isAllowed/8 | Ya | a. Bila <i>endpoint</i> benar maka akan menampilkan status true, data dan status <i>respon</i> 200 OK<br>b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found<br>c. Berikut isi dari JSON:<br><pre>{   "result": true,   "data": "allowed" }</pre> |
|----|---------------------|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### 4.1.1.2. Hasil Pengujian Fungsional Method *Post*

**Tabel 4.9 Pengujian Fungsional pada *Endpoint* Dengan Tipe *Request Post* Mengajukan Donasi Untuk Sebuah *Project***

| No | Skenario Proses | Sistem Bekerja (Ya/Tidak)* | Keterangan                                                                                                                                                                                                                                                                      |
|----|-----------------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | project/donate  | Ya                         | a. Bila <i>endpoint</i> benar maka akan menampilkan data result, ref_code dan status <i>respon</i> 201 Created<br>b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found<br>c. Berikut isi dari JSON:<br><pre>{   "result": true,   "ref_code": "147269" }</pre> |

**Tabel 4.10 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Mengajukan Donasi Untuk Sebuah *Project (Non Member)***

| No | Skenario Proses | Sistem Bekerja (Ya/Tidak)* | Keterangan                                                                                                                                           |
|----|-----------------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | project/donateX | Ya                         | a. Bila <i>endpoint</i> benar maka akan menampilkan data result, ref_code dan status <i>respon</i> 201 Created<br>b. Bila <i>endpoint</i> salah maka |

---

status *respon* 404 Not Found  
 c. Berikut isi dari JSON:  
 {  
   "result": true,  
   "ref\_code": "340359"  
 }

---

**Tabel 4.11 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Melakukan Cek Terhadap Status Donasi**

| No | Skenario Proses      | Sistem Bekerja (Ya/Tidak)* | Keterangan                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----|----------------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | project/statusDonasi | Ya                         | a. Bila <i>endpoint</i> benar maka akan menampilkan data, email, nama_depan, nama_belakang, telp, nominal, donation_status dan status <i>respon</i> 200 OK<br>b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found<br>c. Berikut isi dari JSON:<br>{<br>"result": true,<br>"data": {<br>"email":<br>"nurboy@gmail.com",<br>"nama_depan":<br>"nurmansyah",<br>"nama_belakang":<br>"santosa",<br>"telp": "0812312748",<br>"nominal": 100000,<br>"donation_status": "0"<br>}<br>} |

---





```

E5MDIzZTBmM2IzOGI4Nz
BjMWQxIn0.ntRP7eE78T3Ii8
eeUunc_eAFxPx3FWsdl3aSf-
mqeFE",
 "data": {
 "id_user": 13,
 "nama_depan": "serian",
 "nama_belakang":
"iskandar",
 "email":
"serian@gmail.com",
 "telp": "085691122223",
 "jenis_kelamin": "L",
 "foto":
"http://localhost/wui-
service/storage/app/img/users/
user13.jpg",
 "about": ""
 }
}

```

**Tabel 4.14 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Mendaftarkan User**

| No | Skenario Proses | Sistem Bekerja (Ya/Tidak)* | Keterangan                                                                                                                                                                                                                                    |
|----|-----------------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | daftar          | Ya                         | a. Bila <i>endpoint</i> benar maka akan menampilkan status true dan status <i>respon</i> 201 Created<br>b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found<br>c. Berikut isi dari JSON:<br><pre>{   "result": true }</pre> |

**Tabel 4.15 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Mengirim Ulang Email Verifikasi Akun User**

| No | Skenario Proses | Sistem Bekerja (Ya/Tidak)* | Keterangan                                                                                      |
|----|-----------------|----------------------------|-------------------------------------------------------------------------------------------------|
| 1. | resendEmail     | Ya                         | a. Bila <i>endpoint</i> benar maka akan menampilkan status true dan status <i>respon</i> 200 OK |

- 
- b. Bila *endpoint* salah maka status *respon* 404 Not Found
  - c. Berikut isi dari JSON:
 

```
{
 "result": true
 }
```
- 

**Tabel 4.16 Pengujian Fungsional pada *Endpoint* Dengan *Request Post* Melakukan Reset Password**

| No | Skenario Proses | Sistem Bekerja (Ya/Tidak)* | Keterangan                                                                                                                                                                                                                                                                                                                                                                             |
|----|-----------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | resetPassword   | Ya                         | <ul style="list-style-type: none"> <li>a. Bila <i>endpoint</i> benar maka akan menampilkan status true, data dan status <i>respon</i> 200 OK</li> <li>b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found</li> <li>c. Berikut isi dari JSON:               <pre>{                 "result": true,                 "data": "Sukses"               }</pre> </li> </ul> |

#### 4.1.1.3. Hasil Pengujian Fungsional Method *Put*

**Tabel 4.17 Pengujian Fungsional pada *Endpoint* Dengan *Request Put Update* Nama Depan dan Nama Belakang *User***

| No | Skenario Proses | Sistem Bekerja (Ya/Tidak)* | Keterangan                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----|-----------------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | <i>update</i>   | Ya                         | <ul style="list-style-type: none"> <li>a. Bila <i>endpoint</i> benar maka akan menampilkan data <i>id_user</i>, <i>nama_depan</i>, <i>nama_belakang</i>, <i>telepon</i>, <i>jenis_kelamin</i>, <i>about</i> dan status <i>respon</i> 200 OK</li> <li>b. Bila <i>endpoint</i> salah maka status <i>respon</i> 404 Not Found</li> <li>c. Berikut isi dari JSON:               <pre>{                 "result": true,                 "data": { </pre> </li> </ul> |

```

 "id_user": "13",
 "nama_depan":
 "muhammad",
 "nama_belakang":
 "seria",
 "telepon":
 "08569112223",
 "jenis_kelamin": "L",
 "about": "saya
 mahasiswa ptik nr 2011"
 }
}

```

**Tabel 4.18 Pengujian Fungsional pada *Endpoint Dengan Request Put Ganti Password User***

| No | Skenario Proses | Sistem Bekerja (Ya/Tidak)* | Keterangan                                                                                                                                                                                                                                                                                                                                |
|----|-----------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | editPass        | Ya                         | a. Bila <i>endpoint</i> benar maka akan menampilkan status <i>result true</i> dan status <i>respon 200 OK</i><br>b. Bila <i>endpoint</i> salah maka status <i>respon 404 Not Found</i><br>c. Berikut isi dari JSON:<br><pre>           {             "result": true,             "alert": "Password Changed"           }           </pre> |

**Tabel 4.19 Pengujian Fungsional pada *Endpoint Dengan Request Put Upload Foto User***

| No | Skenario Proses  | Sistem Bekerja (Ya/Tidak)* | Keterangan                                                                                                                                                                                                                                                                                  |
|----|------------------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | uploadGambarUser | Ya                         | a. Bila <i>endpoint</i> benar maka akan menampilkan data foto dan status <i>respon 200 OK</i><br>b. Bila <i>endpoint</i> salah maka status <i>respon 404 Not Found</i><br>c. Berikut isi dari JSON:<br><pre>           {             "result": true,             "data": {           </pre> |

---

```
"foto": "http://localhost/
wui-
service/storage/app/img/user
s/user13.jpg"
}
}
```

---

## 4.2. Pembahasan

### 4.2.1. Pembahasan Hasil Pengujian Fungsional

Berdasarkan Tabel pengujian pada Tabel 4.1 hingga 4.19 yang dievaluasi oleh penguji ahli, penguji ahli adalah seorang programmer *backend* yang bekerja di perusahaan Jakarta *Smart City* alumni PTIK *non regular* 2011 yang bernama Akbar Kusuma Negera, S.Pd. Dapat dilihat bahwa hasil pengujian menunjukkan tidak adanya kesalahan atau *error* pada fungsionalitas *service*.

Setiap *endpoint* merepresentasikan hasil yang sesuai permintaan dari *frontend* seperti ditunjukkan pada Tabel 4.1 sampai 4.19 sehingga dapat dipastikan bahwa pengujian fungsional *service* sudah sepenuhnya bekerja sesuai kriteria keberhasilan *web service* pada Tabel 3.13 dan layak digunakan untuk *backend web*.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Hal pertama yang dilakukan untuk membuat sebuah *web service website* penggalangan dana Wake Up Indonesia adalah melakukan observasi. Observasi dilakukan di Sekretariat Wake Up Indonesia. Setelah itu, melakukan wawancara dengan ketua yayasan Wake Up Indonesia yaitu Irfan Hakim terkait dengan pembuatan *website*. *Fitur-fitur* yang ada dalam *website* Wake Up Indonesia dijadikan sebuah *web service*.

Hal selanjutnya yang dilakukan adalah implementasi metodologi. Aplikasi ini dibuat menggunakan metode pengembangan *software Spiral* dengan memperhatikan kebutuhan-kebutuhan fungsional serta non-fungsional sistem menggunakan sebuah *framework*, *laravel framework* untuk *backend*. Setelah *web service* dibuat, *web service* harus diuji terlebih dahulu. Pengujian *web service* menggunakan metode *Unit Testing*. Uji coba sistem dilakukan oleh penguji ahli.

Hasil pengujian menunjukkan bahwa *web service* yang dirancang telah memenuhi spesifikasi yang diharapkan atau sesuai dengan kriteria keberhasilan *web service* pada tabel 3.13.

##### 5.1.1. Implikasi di Bidang Keteknikan

1. Metode, pendekatan, teknologi, serta teknik pemrograman yang digunakan dalam pengembangan *web service* pada skripsi ini dapat dijadikan referensi dalam pengembangan *web service* sejenis.
2. Dapat didemonstrasikan pada perkuliahan sebagai salah satu contoh pengembangan *web service* di bidang Teknik Informatika dan Komputer.

### 5.1.2. Implikasi di Bidang Pendidikan

1. Dapat dijadikan referensi pengembangan *web service* pada penelitian tugas akhir, terutama untuk bidang Teknik Informatika dan Komputer.
2. Teknologi dan teknik pemrograman yang digunakan dapat dijadikan bahan ajar perkuliahan pada mata kuliah tertentu.

### 5.2. Saran

Untuk penelitian dan pengembangan lebih lanjut, disarankan untuk melakukan berikut:

1. Dalam melakukan penginstalan software untuk merancang *web service*, disarankan laptop/pc yang memiliki RAM 3 GB atau lebih agar penginstalan berhasil dan hal ini juga dimaksudkan untuk menghindari loading (beban).
2. Apabila ingin mengembangkan *web service* Wake Up Indonesia, maka disarankan untuk menambahkan fitur-fitur *endpoint* yang belum ada sesuai kebutuhan yang diinginkan pihak Wake Up Indonesia.

## DAFTAR PUSTAKA

- Arief, R. 2011. *Pemrograman Web Dinamis Menggunakan PHP dan MySQL*. Yogyakarta: CV. Andi Offset
- Hall, J. 2011. *Sistem Informasi Akuntansi*. Canada: Cengage Learning South Western.
- Hariyanto, B. 2004. *Sistem Manajemen Database*. Bandung: Informatika
- Pengertian JSON*, <http://www.json.org>. (diakses: 10 Januari 2016).
- Kadir, A. 2003. *Pengenalan Sistem Informasi*. Yogyakarta: Andi
- Kadir, A. 2008: *Dasar Pemrograman Web Dinamis Menggunakan PHP*, Andi Offset
- Kadir, A. 2008. *Tuntunan Praktis Belajar Database Menggunakan MySQL*, C.V Andi Offset. Yogyakarta.
- KBBI. *Penggalangan*. <http://kbbi.web.id/galang>. (diakses 11 Januari 2016).
- KBBI. *Dana*. <http://kbbi.web.id/dana>. (diakses 11 Januari 2016).
- Mulyanto, A. 2009. *Sistem Informasi Konsep dan Aplikasi*. Pustaka Pelajar. Yogyakarta
- Oliviera, L; Rodrigues, J; Elias, A; & Bruno B. 2014. Ubiquitous Monitoring Solution for Wireless Sensor Networks with Push Notifications and End to-End Connectivity. *Mobile Information System* 10:19-35
- Pasquali, S. 2015. *Deploying Node.js*. Pack Publishing Ltd. Birmingham
- Pengertian MySQL database*. <http://www.mysqldevelopment.com> [10 Januari 2016]
- Ramon A, & Pauline K. 2007. *Dasar-Dasar Database Relasional*. Surabaya: Erlangga.
- Septian, G. 2011. *Trik Pintar Menguasai CodeIgniter*, Elex Media Komputindo, Jakarta
- Sugiyono. 2009. *Metode Penelitian Bisnis (Pendekatan Kuantitatif, Kualitatif, dan R&D)*. Bandung: Alfabeta.
- Stephen & Plew. 2000. *Database Design*. USA: Sams Publishing.



W3C. *Web Service*. <https://www.w3.org/TR/2004/NOTE-ws-arch-20040211>.  
(diakses 10 Januari 2016).

Wahli, U., Burroughs, O., Cline, O., Tung, L., *Services Handbook for WebSphere Application Server 6.1*. [diakses 10 Januari 2016] tersedia :  
<http://www.redbooks.ibm.com/redbooks/pdfs/>.

Winarno, E. 2011. *Easy Web Programming with PHP plus HTML 5*. Jakarta: Elex Media Komputindo.

**Lampiran 1. Menampilkan Surat Bahwa Penulis Telah Melakukan Penelitian di Wake Up Indonesia**

**Wake up  
INDONESIA**

**SURAT KETERANGAN TELAH MELAKUKAN PENELITIAN**  
No. 01/Ket/WI/I/16

Yang bertanda tangan dibawah ini :

Nama : Irfan Hakim  
Jabatan : Ketua Gerakan Wake up Indonesia  
Alamat : Jln. Raya Kresek, Duri Kosambi, Cengkareng, Jakarta Barat


Dengan ini menerangkan bahwa mahasiswa yang beridentitas :

Nama : Muhammad Nurmansyah Santosa  
NIM : 5235117090  
Fakultas : Teknik  
Jurusan : Teknik Elektro  
Universitas : Universitas Negeri Jakarta

Telah selesai melakukan penelitian pada bulan September-Desember 2015 pada Gerakan Wake up Indonesia untuk memperoleh data dalam rangka pembuatan web [www.wakeupindonesia.com](http://www.wakeupindonesia.com) sebagai syarat menyelesaikan skripsi.

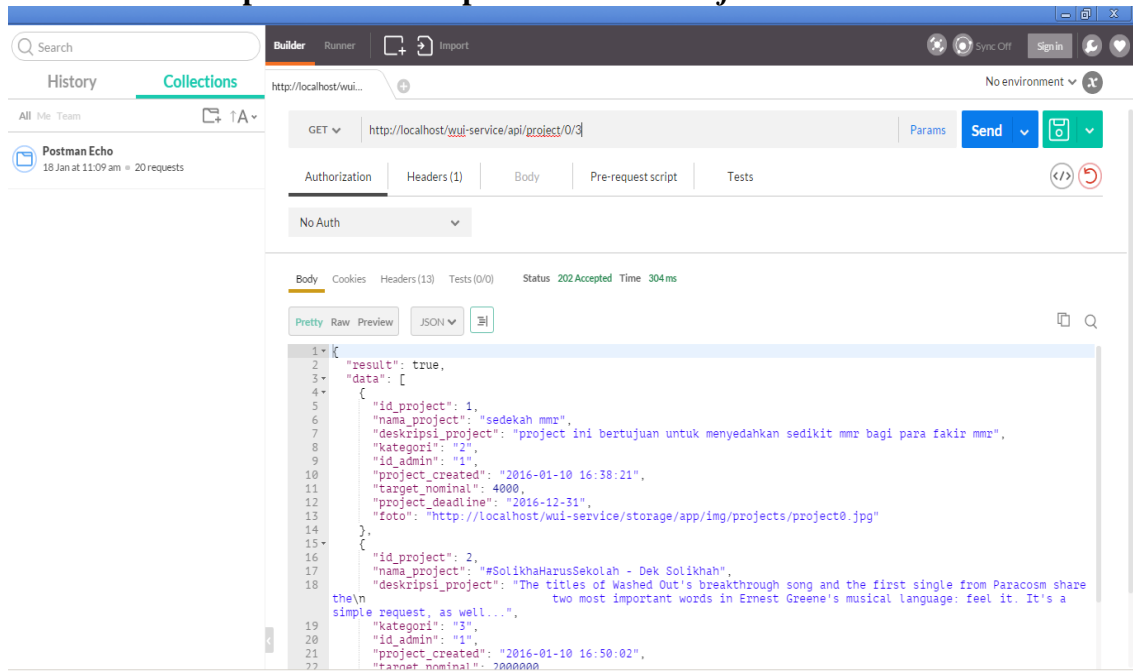
Demikian surat keterangan ini dibuat dan diberikan kepada yang bersangkutan untuk dipergunakan seperlunya.

Jakarta, 18 Januari 2016  
Ketua Gerakan Wake up Indonesia

  
**Irfan Hakim**  
NTA. 0115W/01

**Jalan Raya Kresek, Duri Kosambi, Cengkareng - Jakarta Barat 11750**  
**Telp. 0888 610 1187 - 0812 550 910 80 E-mail. [wakeup.idn@gmail.com](mailto:wakeup.idn@gmail.com)**

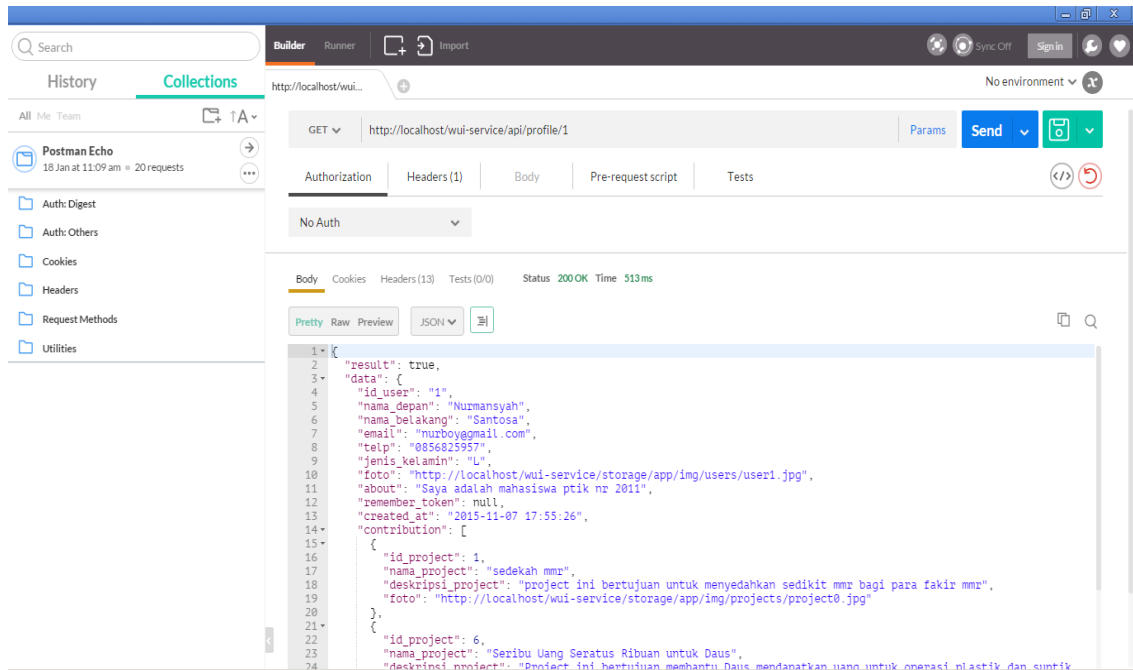
## Lampiran 2. Menampilkan Seluruh *Project* Secara Partial



The screenshot displays the Postman interface for a REST client. The request is a GET call to `http://localhost/wui-service/api/project/0/3`. The response is a 202 Accepted status with a 304ms response time. The response body is shown in JSON format, containing a list of project data.

```
1 {
2 "result": true,
3 "data": [
4 {
5 "id_project": 1,
6 "nama_project": "sedekah mmr",
7 "deskripsi_project": "project ini bertujuan untuk menyedahkan sedikit mmr bagi para fakir mmr",
8 "kategori": "2",
9 "id_admin": "1",
10 "project_created": "2016-01-10 16:38:21",
11 "target_nominal": 4000,
12 "project_deadline": "2016-12-31",
13 "foto": "http://localhost/wui-service/storage/app/img/projects/project0.jpg"
14 },
15 {
16 "id_project": 2,
17 "nama_project": "#SolikhaHarusSekolah - Dek Solikhah",
18 "deskripsi_project": "The titles of Washed Out's breakthrough song and the first single from Paracosm share
19 the\n
20 simple request, as well...",
21 "kategori": "3",
22 "id_admin": "1",
23 "project_created": "2016-01-10 16:50:02",
24 "target_nominal": 2000000
25 }
26]
27 }
```

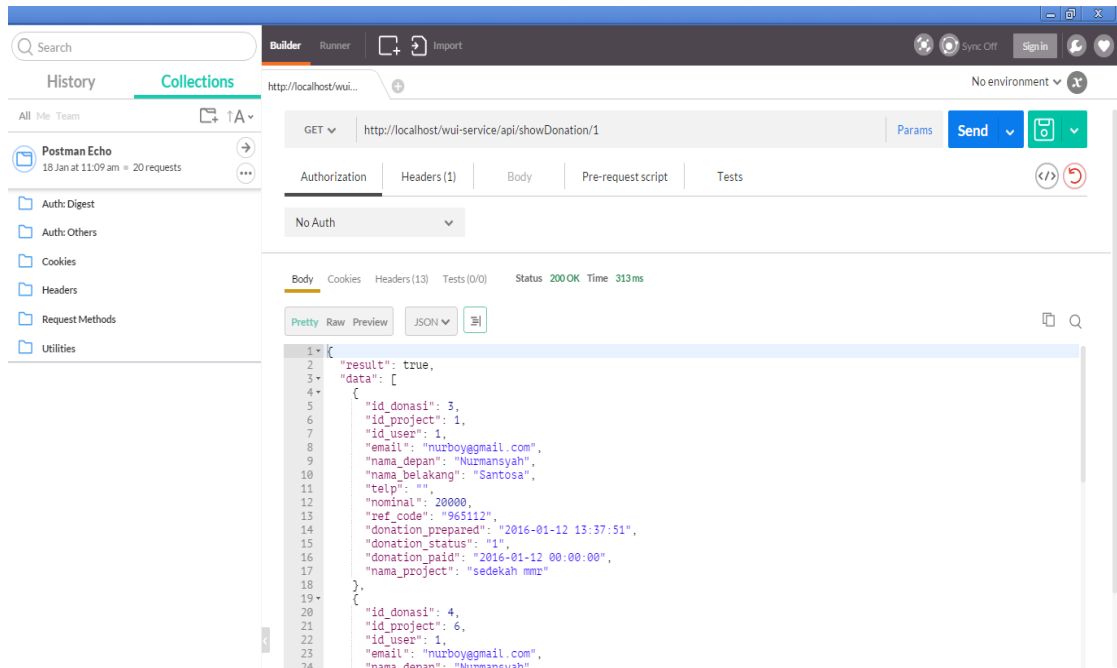
### Lampiran 3. Menampilkan Profil *User*



The screenshot shows the Postman interface with a GET request to `http://localhost/wui-service/api/profile/1` executed successfully. The response is a JSON object with the following structure:

```
1 {
2 "result": true,
3 "data": {
4 "id_user": "1",
5 "nama_depan": "Nuzmansyah",
6 "nama_belakang": "Santosa",
7 "email": "nuzboy@gmail.com",
8 "telp": "0856825957",
9 "jenis_kelamin": "L",
10 "foto": "http://localhost/wui-service/storage/app/img/users/user1.jpg",
11 "about": "Saya adalah mahasiswa ptik nr 2011",
12 "remember_token": null,
13 "created_at": "2015-11-07 17:55:26",
14 "contribution": [
15 {
16 "id_project": 1,
17 "nama_project": "sedekah mmm",
18 "deskripsi_project": "project ini bertujuan untuk menyedahkan sedikit mmm bagi para fakir mmm",
19 "foto": "http://localhost/wui-service/storage/app/img/projects/project0.jpg"
20 },
21 {
22 "id_project": 6,
23 "nama_project": "Seribu Uang Seratus Ribuan untuk Daus",
24 "deskripsi_project": "Project ini bertujuan membantu Daus mendapatkan uang untuk operasi plastik dan suntik
```

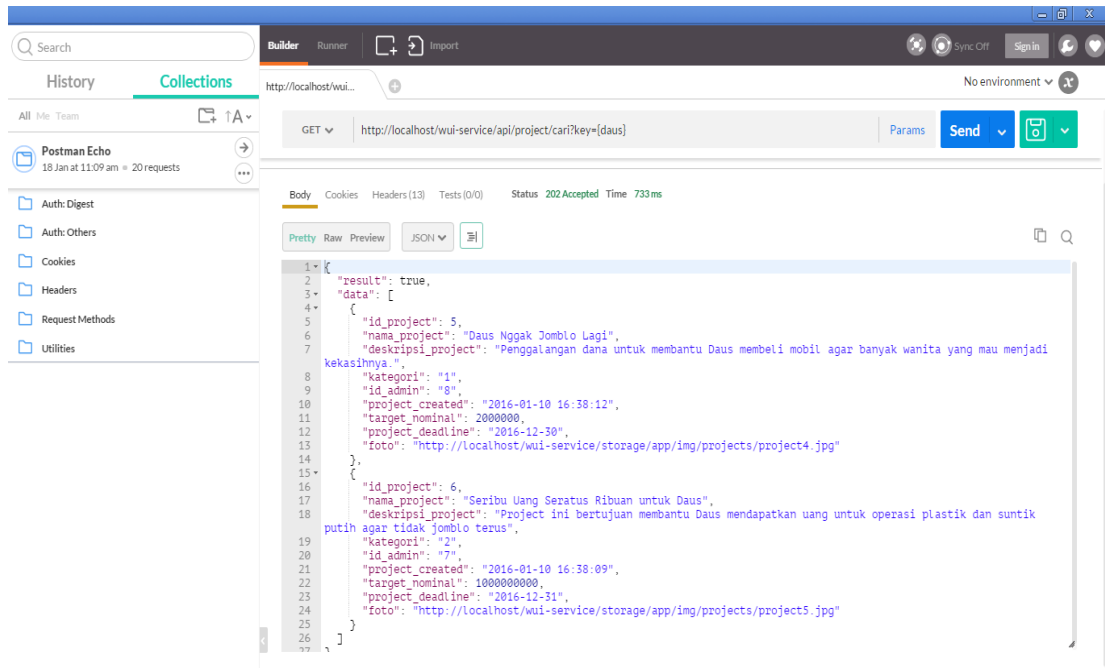
## Lampiran 4. Menampilkan Record Donasi User



The screenshot displays the Postman interface for a REST client. The URL bar shows the endpoint `http://localhost/wui-service/api/showDonation/1`. The request method is `GET`. The response status is `200 OK` with a response time of `313ms`. The response body is displayed in the `Body` tab, showing a JSON array of two donation records.

```
1 {
2 "result": true,
3 "data": [
4 {
5 "id_donasi": 3,
6 "id_project": 1,
7 "id_user": 1,
8 "email": "nurboy@gmail.com",
9 "nama_depan": "Nuzmansyah",
10 "nama_belakang": "Santosa",
11 "telp": "",
12 "nominal": 20000,
13 "ref_code": "965112",
14 "donation_prepared": "2016-01-12 13:37:51",
15 "donation_status": "1",
16 "donation_paid": "2016-01-12 00:00:00",
17 "nama_project": "sedekah mmm"
18 },
19 {
20 "id_donasi": 4,
21 "id_project": 6,
22 "id_user": 1,
23 "email": "nurboy@gmail.com",
24 "nama_depan": "Nuzmansyah"
```

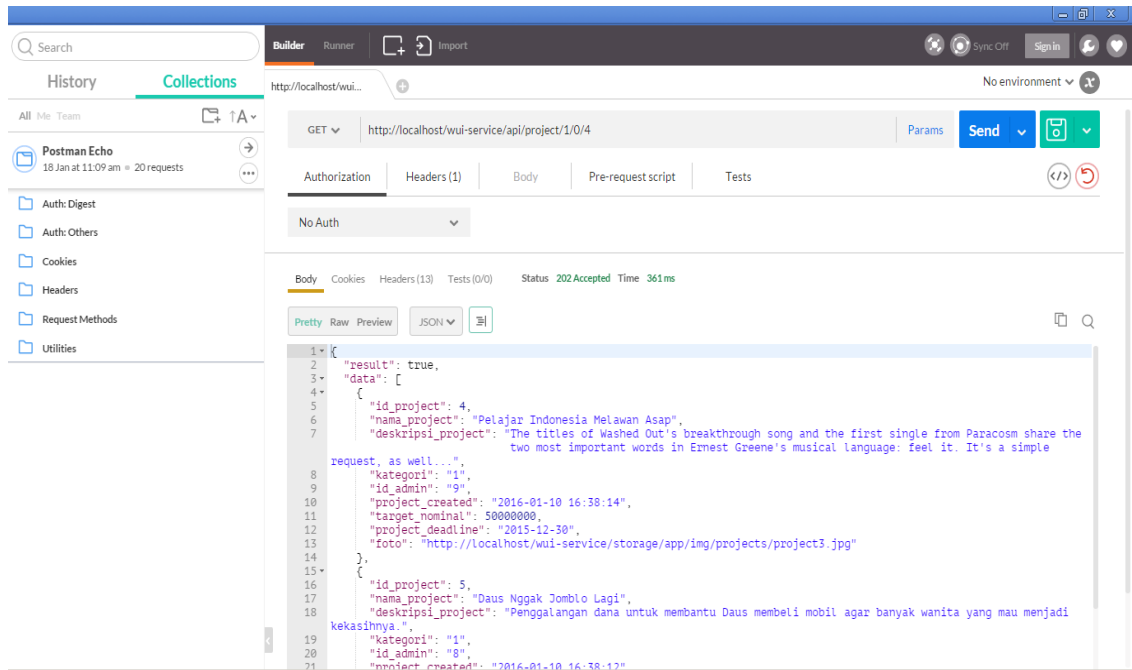
## Lampiran 5. Melakukan Pencarian *Project* Berdasarkan *Keyword*



The screenshot displays the Postman interface. The top bar shows the 'Builder' tab and the URL 'http://localhost/wui...'. The request method is 'GET' and the URL is 'http://localhost/wui-service/api/project/cari?key=[daus]'. The response status is '202 Accepted' with a time of '733 ms'. The response body is shown in 'Pretty' format as a JSON object:

```
1 {
2 "result": true,
3 "data": [
4 {
5 "id_project": 5,
6 "nama_project": "Daus Nggak Jomblo Lagi",
7 "deskripsi_project": "Penggalangan dana untuk membantu Daus membeli mobil agar banyak wanita yang mau menjadi
8 kekasihnya.",
9 "kategori": "1",
10 "id_admin": "9",
11 "project_created": "2016-01-10 16:38:12",
12 "target_nominal": 2000000,
13 "project_deadline": "2016-12-30",
14 "foto": "http://localhost/wui-service/storage/app/img/projects/project4.jpg"
15 },
16 {
17 "id_project": 6,
18 "nama_project": "Seribu Uang Seratus Ribuan untuk Daus",
19 "deskripsi_project": "Project ini bertujuan membantu Daus mendapatkan uang untuk operasi plastik dan suntik
20 putih agar tidak jomblo terus.",
21 "kategori": "2",
22 "id_admin": "7",
23 "project_created": "2016-01-10 16:38:09",
24 "target_nominal": 100000000,
25 "project_deadline": "2016-12-31",
26 "foto": "http://localhost/wui-service/storage/app/img/projects/project5.jpg"
27 }
28]
29 }
```

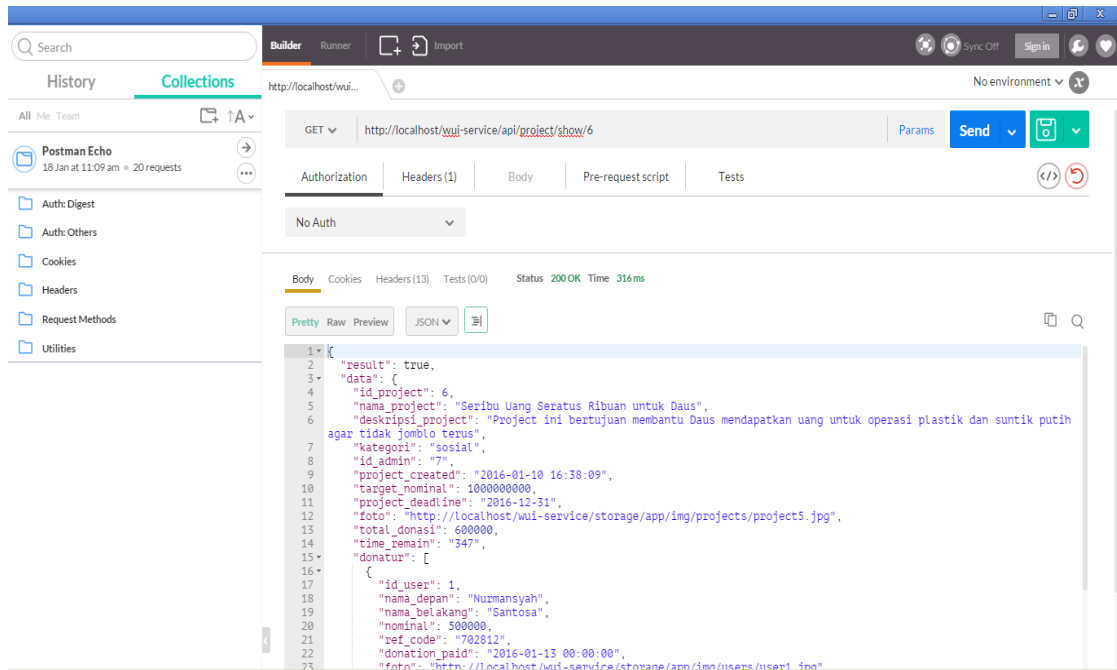
## Lampiran 6. Menampilkan *Project* Berdasarkan Kategori Secara Partial



The screenshot displays the Postman interface for a REST client. The request is a GET call to `http://localhost/wui-service/api/project/1/0/4`. The response is a 202 Accepted status with a 361ms response time. The response body is shown in JSON format, containing a list of two project objects. The first object has an ID of 4, a name 'Pelajar Indonesia Melawan Asap', and a description about Washed Out's song. The second object has an ID of 5, a name 'Daus Nggak Jomblo Lagi', and a description about a fundraising effort.

```
1 {
2 "result": true,
3 "data": [
4 {
5 "id_project": 4,
6 "nama_project": "Pelajar Indonesia Melawan Asap",
7 "deskripsi_project": "The titles of Washed Out's breakthrough song and the first single from Paracosm share the
8 request, as well...",
9 "kategori": "1",
10 "id_admin": "9",
11 "project_created": "2016-01-10 16:38:14",
12 "target_nominal": 50000000,
13 "project_deadline": "2015-12-30",
14 "foto": "http://localhost/wui-service/storage/app/img/projects/project3.jpg"
15 },
16 {
17 "id_project": 5,
18 "nama_project": "Daus Nggak Jomblo Lagi",
19 "deskripsi_project": "Penggalangan dana untuk membantu Daus membeli mobil agar banyak wanita yang mau menjadi
20 kekasihnya.",
21 "kategori": "1",
22 "id_admin": "8",
23 "project_created": "2016-01-10 16:38:12"
24 }
25]
26 }
```

## Lampiran 7. Menampilkan Data Sebuah Project

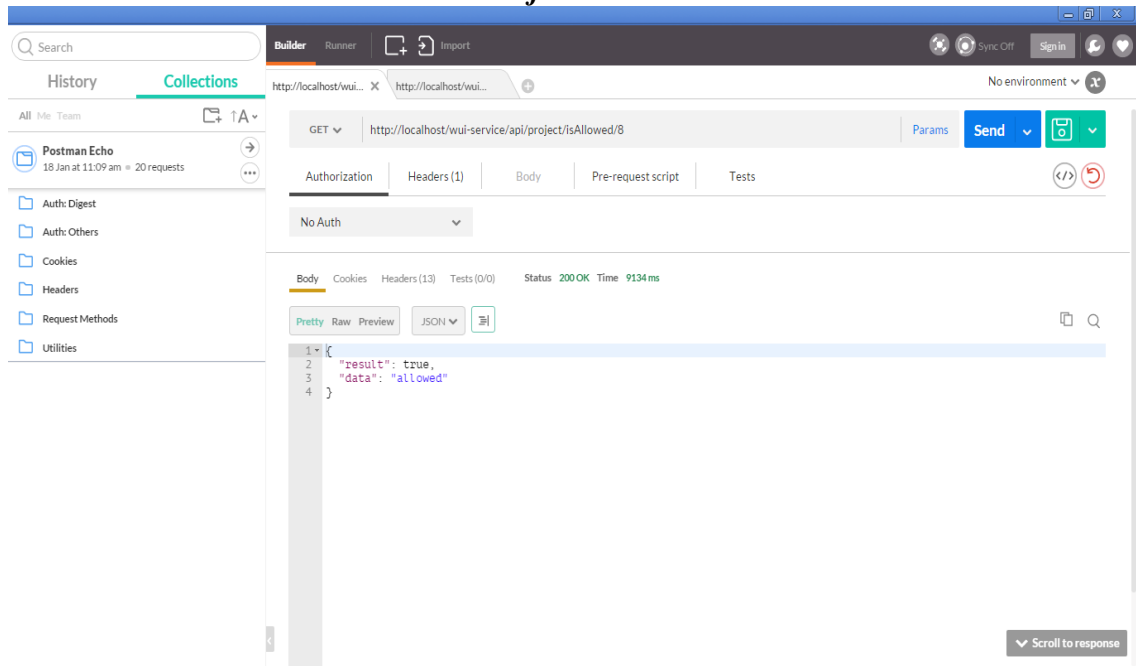


The screenshot displays the Postman interface with a GET request to `http://localhost/wui-service/api/project/show/6` executed successfully. The response is a JSON object with the following structure:

```
1 {
2 "result": true,
3 "data": {
4 "id_project": 6,
5 "nama_project": "Seribu Uang Seratus Ribuan untuk Daus",
6 "deskripsi_project": "Project ini bertujuan membantu Daus mendapatkan uang untuk operasi plastik dan suntik putih
7 agar tidak jomblo terus",
8 "kategori": "sosial",
9 "id_admin": "7",
10 "project_created": "2016-01-10 16:38:09",
11 "target_nominal": 100000000,
12 "project_deadline": "2016-12-31",
13 "foto": "http://localhost/wui-service/storage/app/img/projects/project5.jpg",
14 "total_donasi": 600000,
15 "time_remain": "347",
16 "donatur": [
17 {
18 "id_user": 1,
19 "nama_depan": "Nuzmansyah",
20 "nama_belakang": "Santosa",
21 "nominal": 500000,
22 "ref_code": "702812",
23 "donation_paid": "2016-01-13 00:00:00",
24 "foto": "http://localhost/wui-service/storage/app/img/users/user1.png"
25 }
26]
27 }
28 }
```



## Lampiran 8. Melakukan Pengecekan Apakah Masih Dapat Melakukan Donasi Pada Sebuah *Project*



The screenshot displays the Postman interface. The top navigation bar includes 'Builder', 'Runner', and 'Import' options. The main workspace shows a GET request to the URL `http://localhost/wui-service/api/project/isAllowed/8`. The request is configured with 'No Auth' and is ready to be sent. The response is displayed in the 'Body' tab, showing a JSON object with the following structure:

```
1 {
2 "result": true,
3 "data": "allowed"
4 }
```

The status bar at the bottom indicates a successful response with a status of 200 OK and a response time of 9134 ms. A 'Scroll to response' button is visible in the bottom right corner.

## Lampiran 9. Mengajukan Donasi Untuk Sebuah *Project*

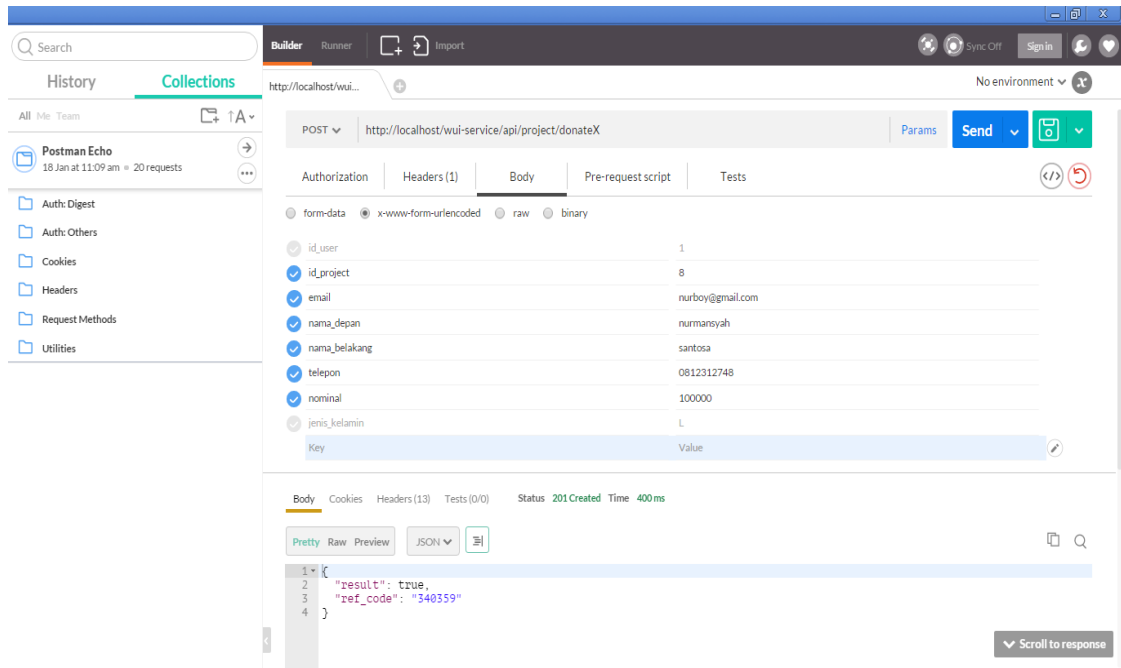
The screenshot displays the Postman interface for a REST client. The main window shows a POST request to the endpoint `http://localhost/wui-service/api/project/donate`. The request body is configured as `x-www-form-urlencoded` and contains the following data:

| Key                            | Value                                    |
|--------------------------------|------------------------------------------|
| <code>id_user</code>           | <code>ZKZKZK</code>                      |
| <code>verified_password</code> | <code>ZKZKZK</code>                      |
| <code>id_user</code>           | <code>1</code>                           |
| <code>id_project</code>        | <code>7</code>                           |
| <code>jenis_kelamin</code>     | <code>L</code>                           |
| <code>about</code>             | <code>saya mahasiswa ptik nr 2011</code> |
| <code>nominal</code>           | <code>200000</code>                      |

The response is displayed in the bottom panel, showing a status of `201 Created` and a time of `624 ms`. The response body is:

```
1 {
2 "result": true,
3 "ref_code": "147269"
4 }
```

## Lampiran 10. Mengajukan Donasi Untuk Sebuah *Project* (Non Member)



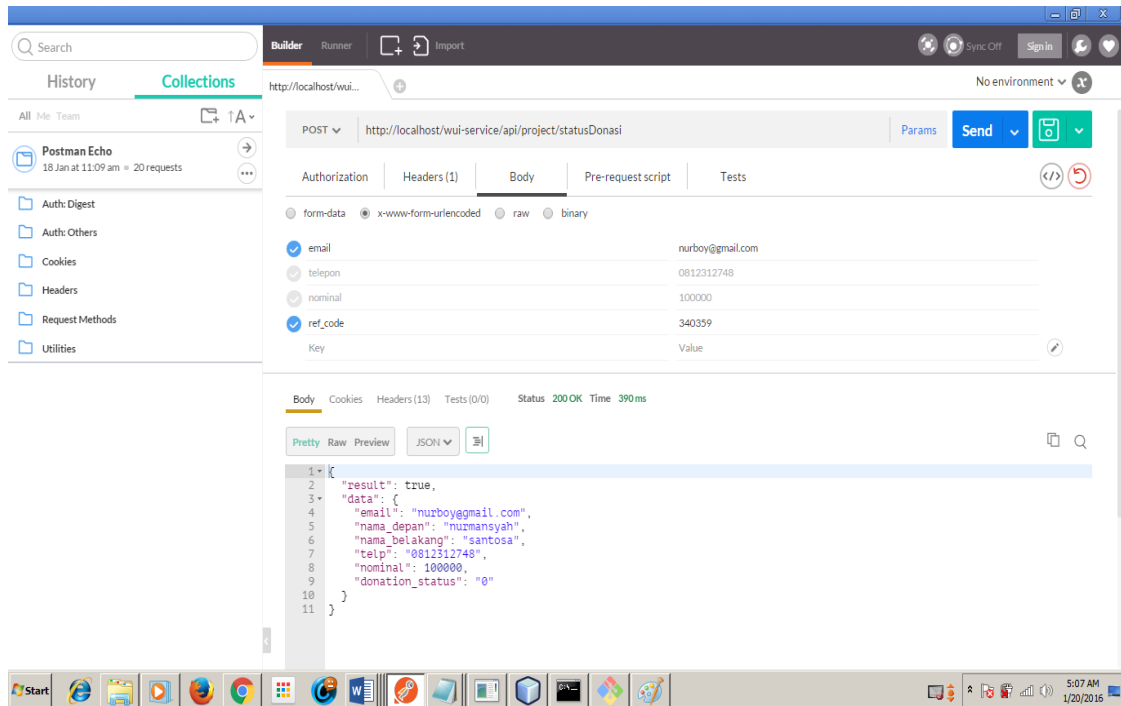
The screenshot displays the Postman interface for a POST request. The URL is `http://localhost/wui-service/api/project/donateX`. The request body is set to `x-www-form-urlencoded` and contains the following data:

| Key                        | Value                         |
|----------------------------|-------------------------------|
| <code>id_user</code>       | <code>1</code>                |
| <code>id_project</code>    | <code>8</code>                |
| <code>email</code>         | <code>nurboy@gmail.com</code> |
| <code>nama_depan</code>    | <code>nummansyah</code>       |
| <code>nama_belakang</code> | <code>santosa</code>          |
| <code>telepon</code>       | <code>0812312748</code>       |
| <code>nominal</code>       | <code>100000</code>           |
| <code>jenis_kelamin</code> | <code>L</code>                |

The response is shown in the bottom panel, with a status of 201 and a time of 400 ms. The response body is a JSON object:

```
1 {
2 "result": true,
3 "ref_code": "340359"
4 }
```

## Lampiran 11. Melakukan Cek Terhadap Status Donasi



The screenshot displays the Postman interface for a REST client. The URL bar shows a POST request to `http://localhost/wui-service/api/project/statusDonasi`. The request body is configured as `x-www-form-urlencoded` with the following parameters:

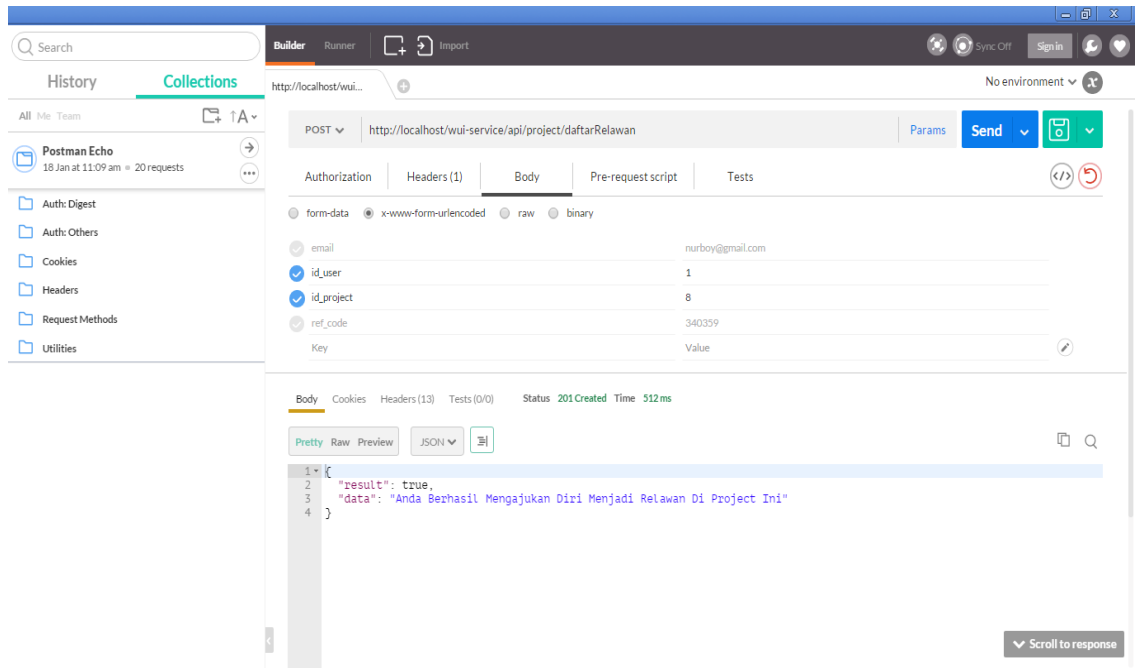
| Key      | Value            |
|----------|------------------|
| email    | nurboy@gmail.com |
| telepon  | 0812312748       |
| nominal  | 100000           |
| ref_code | 340359           |

The response body is shown in JSON format, indicating a successful status check:

```
1 {
2 "result": true,
3 "data": {
4 "email": "nurboy@gmail.com",
5 "nama_depan": "nuzmansyah",
6 "nama_belakang": "santosa",
7 "telp": "0812312748",
8 "nominal": 100000,
9 "donation_status": "0"
10 }
11 }
```

The status bar at the bottom of the interface shows `Status 200 OK Time 390 ms`. The Windows taskbar at the bottom indicates the system time is 5:07 AM on 1/20/2016.

## Lampiran 12. Mengajukan Diri Sebagai Relawan Dalam *Project*



The screenshot displays the Postman interface for a REST client. The main window shows a POST request to the endpoint `http://localhost/wui-service/api/project/daftarRelawan`. The request body is configured as `x-www-form-urlencoded` with the following parameters:

| Key        | Value            |
|------------|------------------|
| email      | nurboy@gmail.com |
| id_user    | 1                |
| id_project | 8                |
| ref_code   | 340359           |

The response is shown in the bottom pane, indicating a `201 Created` status with a response time of `512ms`. The response body is displayed in JSON format:

```
1 {
2 "result": true,
3 "data": "Anda Berhasil Mengajukan Diri Menjadi Relawan Di Project Ini"
4 }
```



## Lampiran 14. Mendaftarkan *User*

The screenshot displays the Postman interface for a REST client. The main window shows a POST request to the endpoint `http://localhost/wui-service/api/daftar`. The request body is formatted as x-www-form-urlencoded and contains the following data:

| Key           | Value               |
|---------------|---------------------|
| nama_depan    | septiani            |
| nama_belakang | iskandar            |
| email         | sepetiani@gmail.com |
| password      | asdfasdf            |
| telepon       | 086710285929        |
| jenis_kelamin | P                   |

The response is shown in the bottom pane, indicating a successful status (201 Created) with a response time of 9224 ms. The response body is a JSON object:

```
1 {
2 "result": true
3 }
```

The Windows taskbar at the bottom shows the system clock at 7:41 AM on 1/20/2016.

## Lampiran 15. Mengirim Ulang Email Verifikasi Akun *User*

The screenshot displays the Postman interface for a REST client request. The request is a POST to `http://localhost/wui-service/api/resendEmail`. The body is set to `x-www-form-urlencoded` and contains the following form data:

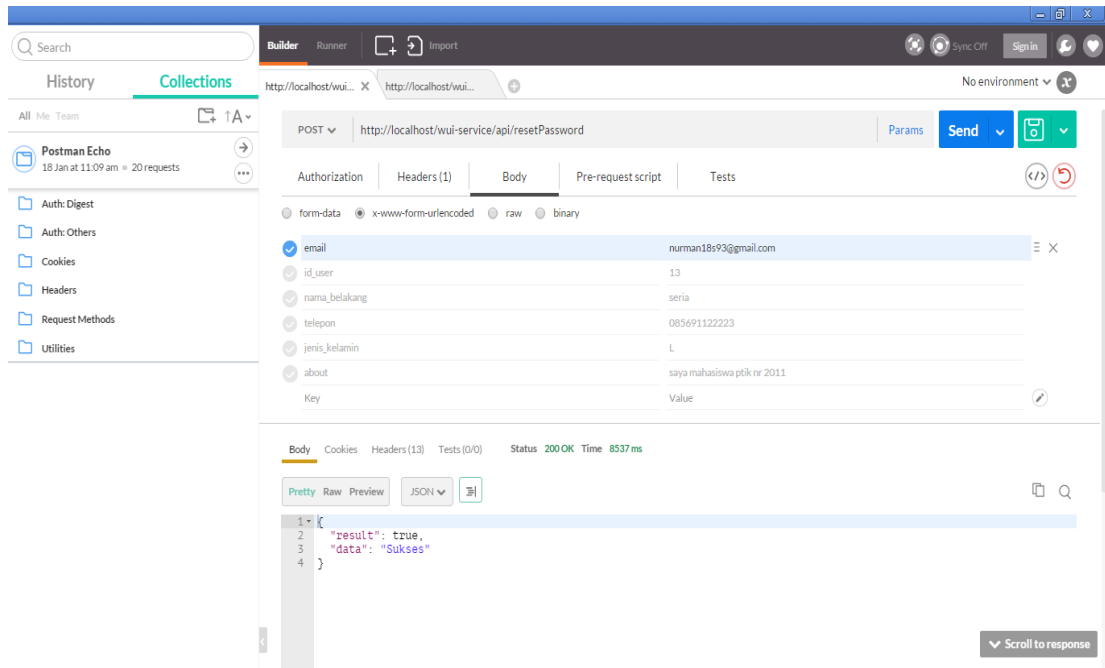
| Key           | Value              |
|---------------|--------------------|
| nama_depan    | septiani           |
| nama_belakang | iskandar           |
| email         | septiani@gmail.com |
| password      | asdfasdf           |
| telepon       | 086710285929       |
| jenis_kelamin | P                  |

The response is shown in the bottom pane, indicating a successful status of 200 OK with a response time of 5179 ms. The response body is a JSON object:

```
1 {
2 "result": true
3 }
```



## Lampiran 16. Melakukan Reset password



The screenshot displays the Postman interface for a REST client. The main window shows a POST request to the endpoint `http://localhost/wui-service/api/resetPassword`. The request body is configured as `x-www-form-urlencoded` and contains the following data:

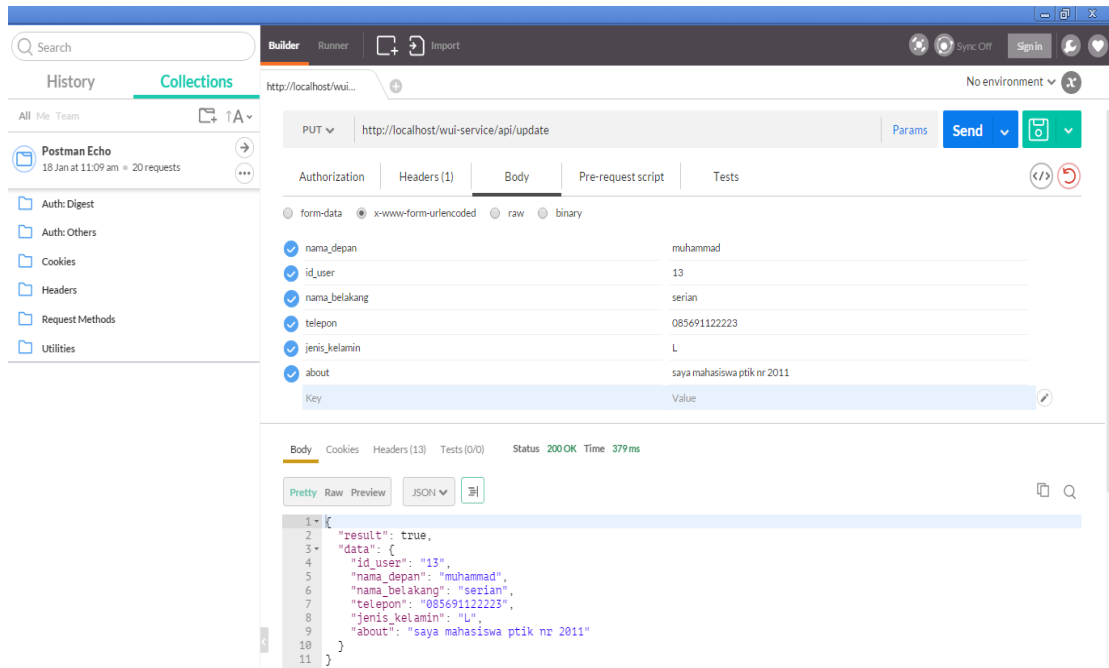
| Key           | Value                       |
|---------------|-----------------------------|
| email         | nurman18s93@gmail.com       |
| id_user       | 13                          |
| nama_belakang | seria                       |
| telepon       | 085691122223                |
| jenis_kelamin | L                           |
| about         | saya mahasiswa ptik nr 2011 |

The response status is `200 OK` with a response time of `8537 ms`. The response body is displayed in JSON format:

```
1 {
2 "result": true,
3 "data": "Sukses"
4 }
```

The interface also shows a sidebar with a 'Collections' view containing folders for 'Auth: Digest', 'Auth: Others', 'Cookies', 'Headers', 'Request Methods', and 'Utilities'. The top navigation bar includes 'Builder', 'Runner', and 'Import' options.

## Lampiran 17. Update Nama Depan dan Nama Belakang User



The screenshot displays the Postman interface for a PUT request to the endpoint `http://localhost/wui-service/api/update`. The request body is a JSON object with the following fields:

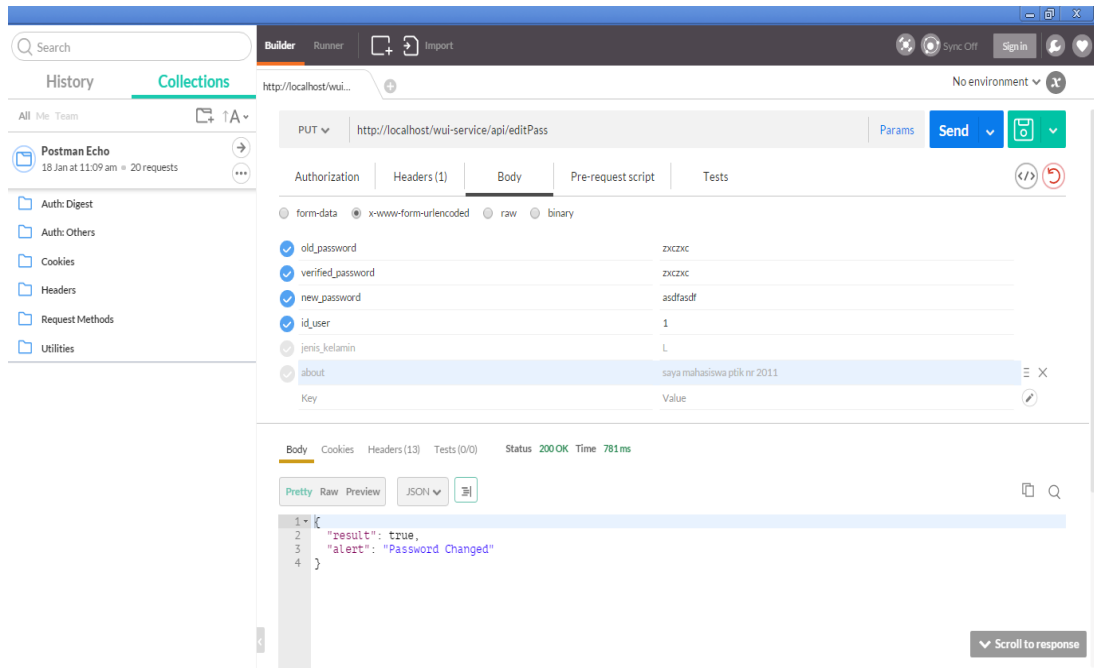
| Key                        | Value                                    |
|----------------------------|------------------------------------------|
| <code>nama_depan</code>    | <code>muhammad</code>                    |
| <code>id_user</code>       | <code>13</code>                          |
| <code>nama_belakang</code> | <code>serian</code>                      |
| <code>telepon</code>       | <code>085691122223</code>                |
| <code>jenis_kelamin</code> | <code>L</code>                           |
| <code>about</code>         | <code>saya mahasiswa ptik nr 2011</code> |

The response is a JSON object with the following structure:

```
1 {
2 "result": true,
3 "data": {
4 "id_user": "13",
5 "nama_depan": "muhammad",
6 "nama_belakang": "serian",
7 "telepon": "085691122223",
8 "jenis_kelamin": "L",
9 "about": "saya mahasiswa ptik nr 2011"
10 }
11 }
```

The status of the request is 200 OK, and the response time is 379 ms.

## Lampiran 18. Ganti Password User



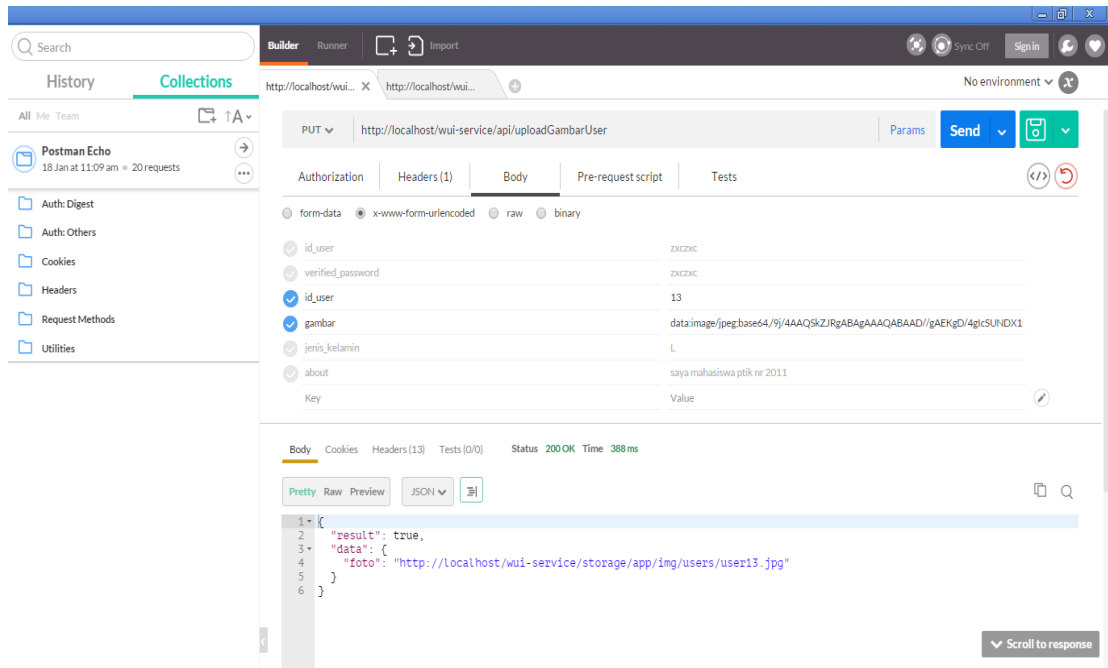
The screenshot displays the Postman interface for a PUT request to `http://localhost/wui-service/api/editPass`. The request body is set to `x-www-form-urlencoded` and contains the following data:

| Key               | Value                       |
|-------------------|-----------------------------|
| old_password      | zxczxc                      |
| verified_password | zxczxc                      |
| new_password      | asdfasdf                    |
| id_user           | 1                           |
| jenis_kelamin     | L                           |
| about             | saya mahasiswa ptik nr 2011 |

The response is shown in the bottom panel, with a status of `200 OK` and a time of `781 ms`. The response body is:

```
1 {
2 "result": true,
3 "alert": "Password Changed"
4 }
```

## Lampiran 19. Upload Foto *User*



The screenshot displays the Postman interface for a PUT request to the endpoint `http://localhost/wui-service/api/uploadGambarUser`. The request is configured with the following parameters:

| Key                            | Value                                                                           |
|--------------------------------|---------------------------------------------------------------------------------|
| <code>id_user</code>           | <code>zxczxc</code>                                                             |
| <code>verified_password</code> | <code>zxczxc</code>                                                             |
| <code>id_user</code>           | <code>13</code>                                                                 |
| <code>gambar</code>            | <code>data:image/jpeg;base64/9j/4AAQSkZJRgABAgAAQABAAD/gAEKgD/4glcSUNDX1</code> |
| <code>jenis_kelamin</code>     | <code>L</code>                                                                  |
| <code>about</code>             | <code>saya mahasiswa ptik nr 2011</code>                                        |

The response is a JSON object with the following structure:

```
1 {
2 "result": true,
3 "data": {
4 "foto": "http://localhost/wui-service/storage/app/img/users/user13.jpg"
5 }
6 }
```

The status of the request is `200 OK` with a response time of `388 ms`. The response body is displayed in a pretty-printed JSON format.

## DAFTAR RIWAYAT HIDUP



Penulis bernama lengkap Muhammad Nurmansyah Santosa, lahir di Bogor 18 September 1993, merupakan anak ke-empat dari pasangan Endang Iskandar dan Sunarsih. Penulis menempuh pendidikan formalnya di SDN Sindang Sari 1, SMP Al-Ghazaly Bogor, dan MAN 2 Bogor.

Pada tahun 2011, penulis diterima sebagai mahasiswa Program Studi Pendidikan Teknik Informatika dan Komputer, Jurusan Teknik Elektro, Fakultas Teknik Universitas Negeri Jakarta melalui jalur mandiri (PENMABA UNJ 2011). Dalam menyelesaikan studinya, penulis mengadakan sebuah penelitian untuk pengerjaan skripsi dengan judul “Pembuatan *Web Service* Untuk *Website* Penggalangan Dana Wake Up Indoensia” sebagai syarat dalam mendapat gelar sarjana pendidikan.