

PENGEMBANGAN DAN *MAINTENANCE* APLIKASI
KESEHATAN PADA PT. GLOBAL URBAN ESENSIAL

Tugas Akhir

Diajukan untuk Memenuhi Salah Satu Persyaratan Mencapai Derajat Sarjana
Informatika



Dibuat Oleh:

Hendrikus Adi Purnama

160709007

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ATMA JAYA YOGYAKARTA
2020

HALAMAN PENGESAHAN

Tugas Akhir Berjudul

PENGEMBANGAN DAN MAINTENANCE APLIKASI KESEHATAN PADA PT. GLOBAL
URBAN ESENSIAL

yang disusun oleh

HENDRIKUS ADI PURNAMA

160709007

dinyatakan telah memenuhi syarat pada tanggal 08 Juli 2020

		Keterangan
Dosen Pembimbing 1	: Dr.Alb. Joko Santoso, MT.	Telah menyetujui
Dosen Pembimbing 2	: Dr. Andi Wahyu Rahardjo, BSEE., MSSE	Telah menyetujui
Tim Penguji		
Penguji 1	: Dr.Alb. Joko Santoso, MT.	Telah menyetujui
Penguji 2	: Paulus Mudjihartono, ST., MT., PhD	Telah menyetujui
Penguji 3	: Vinindita Citrayasa, S.Pd., M.Hum	Telah menyetujui

Yogyakarta, 08 Juli 2020

Universitas Atma Jaya Yogyakarta

Fakultas Teknologi Industri

Dekan

ttd

Dr. A. Teguh Siswanto, M.Sc



PERNYATAAN ORISINALITAS & PUBLIKASI ILMIAH

Saya yang bertanda tangan di bawah ini:

Nama Lengkap : Hendrikus Adi Purnama
NPM : 160709007
Program Studi : Informatika
Fakultas : Teknologi Industri
Judul Penelitian : Pengembangan dan *Maintenance* Aplikasi Kesehatan Pada PT. Global Urban Esensial

Menyatakan dengan ini:

1. Tugas Akhir ini adalah benar tidak merupakan salinan sebagian atau keseluruhan dari karya penelitian lain.
2. Memberikan kepada Universitas Atma Jaya Yogyakarta atas penelitian ini, berupa Hak untuk menyimpan, mengelola, mendistribusikan, dan menampilkan hasil penelitian selama tetap mencantumkan nama penulis.
3. Bersedia menanggung secara pribadi segala bentuk tuntutan hukum atas pelanggaran Hak Cipta dalam pembuatan Tugas Akhir ini.

Demikianlah pernyataan ini dibuat dan dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 8 Juni 2020

Yang menyatakan,

Hendrikus Adi Purnama

160709007

PERNYATAAN PERSETUJUAN DARI INSTANSI ASAL PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama Lengkap Pembimbing : Richard Trisnosaputra

Jabatan : Software Engineer

Departemen : ITGS

Menyatakan dengan ini:

Nama Lengkap : Hendrikus Adi Purnama

NPM : 160709007

Program Studi : Informatika

Fakultas : Teknologi Industri

Judul Penelitian : Pengembangan dan *Maintenance* Aplikasi
Kesehatan Pada PT. Global Urban Esensial

1. Penelitian telah selesai dilaksanakan pada perusahaan.
 2. Perusahaan telah melakukan sidang internal berupa kelayakan penelitian ini dan akan mencantumkan lembar penilaian secara tertutup kepada pihak universitas sebagai bagian dari nilai akhir mahasiswa.
 3. Memberikan kepada Instansi Penelitian dan Universitas Atma Jaya Yogyakarta atas penelitian ini, berupa hak untuk menyimpan, mengelola, mendistribusikan, dan menampilkan hasil penelitian selama tetap mencantumkan nama penulis.
- Demikianlah pernyataan ini dibuat dan dapat dipergunakan sebagaimana mestinya.

Tangerang, Juni 2020

Yang menyatakan,

Richard Trisnosaputra

Software Engineer ITGS

HALAMAN PERSEMBAHAN

Kepada siapapun yang membacanya:

“Ratusan bahkan ribuan lembar skripsi sekalipun tidak akan dapat membalas jerih payah orang tua. Tapi dengan bekal ilmu, gelar, dan kesuksesan akan menjadi jalan untuk membanggakan orang tua”



“Terimakasih buat Yogyakarta, kau memberikan kenangan yang membuatku betah untuk berlama-lama disini, atas masa perkuliahan dan lika-liku perjalananya, kota yang sewaktu saya datang ditemani oleh seseorang yang sempat menjadi spesial di kehidupan saya waktu SMA, tempatku menyembuhkan luka, kisah kasih cinta dengan beberapa orang yang sempat mewarnai perkuliahan saya. Pahit, manis, asam, kecut yang saya rasakan tetapi percayalah akan ada yang selalu kurindukan disini, entah itu tempat, teman, kenangan, dan juga mantan gebetan. Tapi saya percaya selalu ada sesuatu di Jogja”

-Yogyakarta, 12 Juni 2020 -

KATA PENGANTAR

Puji dan syukur penulis haturkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan pembuatan tugas akhir “Pengembangan dan *Maintenance* Aplikasi Kesehatan Pada PT. Global Urban Esensial” ini dengan baik.

Penyusunan tugas akhir ini bertujuan untuk memenuhi salah satu ketentuan buat menggapai derajat sarjana Informatika dari Program Studi Informatika, Fakultas Teknologi Industri di Universitas Atma Jaya Yogyakarta.

Penulis menyadari jika dalam pembuatan tugas akhir ini penulis sudah memperoleh dukungan, bimbingan, serta dorongan dari banyak pihak. Untuk itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Tuhan Yesus Kristus yang menemani saya dalam tiap pengerjaan tugas akhir ini melalui berkat, rahmat, serta karunia-Nya.
2. Bapak Dr. A. Teguh Siswanto, M.Sc., selaku Dekan Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta.
3. Bapak Dr. Ir. Alb. Joko Santoso, M.T., selaku dosen pembimbing I yang turut serta selalu membimbing dan mendukung penulis untuk tetap memberikan motivasi dan semangat dalam menyelesaikan tugas akhir ini.
4. Bapak Dr. Andi Wahyu Rahardjo Emanuel, BSEE., MSSE., selaku dosen pembimbing II yang turut serta selalu membimbing dan mendukung penulis untuk tetap memberikan motivasi dan semangat dalam menyelesaikan tugas akhir ini.
5. Seluruh Dosen dan Staff Pengajar Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta yang telah membimbing dan membantu penulis dalam masa perkuliahan yang penulis telah tempuh.
6. Orang tua (Karel Yoseph Rauy dan AG Heny Susilowati), adik (Ignatius Chris Prasetyo dan Rafael Vitto Matarauy) dan keluarga besar yang mendukung serta memberikan doa dan semangat kepada penulis sehingga tugas akhir ini dapat

selesai dengan tepat waktu.

7. Bapak Ferry Soetikno, selaku Presiden Direktur PT. Dexe Medica yang memberikan kesempatan kepada penulis agar dapat melakukan program magang di PT. Global Urban Esensial.
8. Mas Rahmat Hidayat, Mas Resa, dan Kak Richard, selaku mentor program magang pada PT. Global Urban Esensial yang selalu memberikan bimbingan dan semangat agar proyek magang dapat terlaksana dengan baik dan lancar.
9. Sahabat dari GP Netco 13 yaitu Ricky Erianto Sianipar, Edoa, Heri, Josua, Krisfanri, Pandu, Samuel, Steven dan Vincent yang sudah bersedia menemani dan mendengar keluh-kesah selama penulis menjalani masa perkuliahan ini.
10. Teman-teman magang *Dexe Group* berdua belas: Octo, Hilton, Yulita, Listra, Dewa, Tovid, Mei, Yoana, Panda, Marsel dan Dedi.
11. Sahabat Perkuliahan yaitu Hakeem Nanda Saputra, Surya Jaya Raka, Marcell Pascananda, Yudho, Jose, Yafet, Vian, dan semua teman yang sudah memberikan banyak motivasi dan bantuan bagi penulis selama masa perkuliahan dan masa penulisan Tugas Akhir ini.
12. Brigitta Olivia Liestanto Putri yang telah menjadi salah satu orang terdekat yang spesial bagi penulis.
13. Teman-teman P3L yaitu Rizka dan Wilson atas semangat berjuang dengan penulis selama menempuh mata kuliah P3L.
14. Teman-teman Asisten Dosen Basis Data dan Asisten Dosen Administrasi Basis Data 2019, yang telah menjadi keluarga baru dan tempat berbagi dalam mengembangkan soft skill.
15. Teman-teman KKN 75 Unit R kelompok 75 yang telah memberikan cerita yang mengesankan kepada penulis.

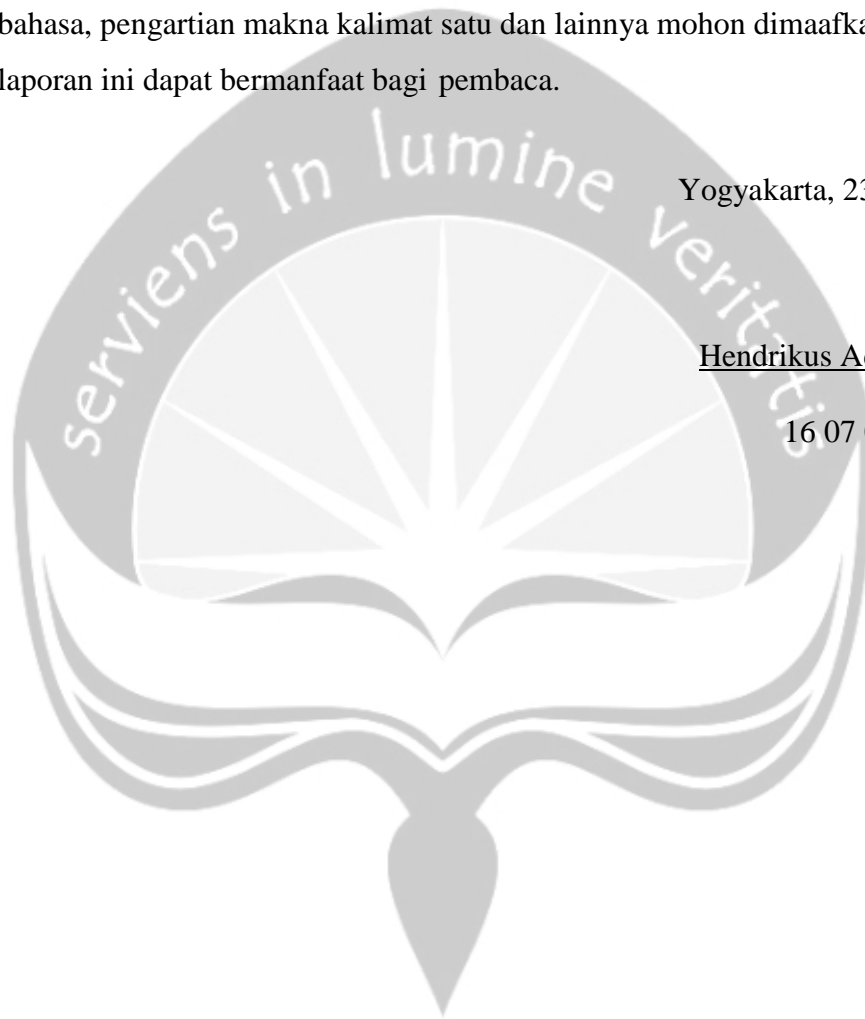
- 16 Seluruh teman-teman yang mendukung penulis dan tidak dapat disebutkan satu per satu yang tentunya telah memberikan semangat dan kasih sayang kepada penulis.

Pembuatan projek magang dan juga penulisan tugas akhir tentunya dibuat dengan sepenuh hati. Namun tidak dipungkiri bahwa dalam proses penulisan tugas akhir ini terdapat banyak kekurangan. Maka dari itu, apabila pembaca menemukan kekurangan dan kekeliruan dalam hal penulisan, penggunaan bahasa, pengartian makna kalimat satu dan lainnya mohon dimaafkan. Semoga laporan ini dapat bermanfaat bagi pembaca.

Yogyakarta, 23 Juni 2020

Hendrikus Adi Purnama

16 07 09007



DAFTAR ISI

PENGEMBANGAN DAN <i>MAINTENANCE</i> APLIKASI KESEHATAN PADA PT. GLOBAL URBAN ESENSIAL	i
HALAMAN PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS & PUBLIKASI ILMIAH.....	i
PERNYATAAN PERSETUJUAN DARI INSTANSI ASAL PENELITIAN	ii
HALAMAN PERSEMBAHAN	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	x
INTISARI.....	x
BAB I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian	1
1.5 Metode Penelitian.....	1
a. Studi Literatur	1
b. Analisis.....	1
c. Implementasi Perangkat Lunak.....	1
d. Pengujian Perangkat Lunak.....	1
e. Pelaporan.....	1
1.6. Sistematika Penulisan	4
BAB II. TINJAUAN PUSTAKA.....	5
BAB III. LANDASAN TEORI.....	11
3.1. <i>Microservice</i>	11

3.2.	<i>Content Management System</i>	12
3.3.	JavaScript	12
3.4.	Node.js	13
3.5.	Express.js	13
3.6.	MySQL.....	14
3.7.	Next.js	15
3.8.	Angular.....	15
BAB IV. ANALISIS DAN PERANCANGAN SISTEM		17
4.1	Analisis Sistem.....	17
4.2	Lingkup Masalah.....	18
4.3	Perspektif Produk	19
4.4	Fungsi Produk	19
4.5	Kebutuhan Antarmuka	35
4.6	Perancangan	37
4.6.1.	Perancangan Arsitektur	37
4.6.2.	Perancangan Antarmuka	50
BAB V. IMPLEMENTASI DAN PENGUJIAN SISTEM.....		72
5.1	Implementasi Sistem Implementasi Antarmuka	72
5.1.1	Halaman Tambah Data CMS (<i>Content Management System</i>).....	72
5.1.2	Halaman <i>Edit</i> Data CMS (<i>Content Management System</i>)	75
5.1.3	Halaman Hapus Data CMS (<i>Content Management System</i>).....	78
5.1.4	Halaman Login Untuk CMS (<i>Content Management System</i>)	81
5.1.5	Halaman Cari dan Tampil Data Tipe	83
5.1.6	Halaman Tambah Data Tipe	85
5.1.7	Halaman <i>Edit</i> Data Tipe.....	87
5.1.8	Halaman Hapus Data Tipe	89

5.1.9	Halaman Cari dan Tampil Data Komponen.....	91
5.1.10	Halaman Tambah Data Komponen.....	93
5.1.11	Halaman <i>Edit</i> Data Komponen	96
5.1.12	Halaman Hapus Data Komponen.....	99
5.1.13	Halaman Cari dan Tampil <i>Screen</i>	101
5.1.15	Halaman <i>Edit</i> Data <i>Screen</i>	105
5.1.16	Halaman Hapus Data <i>Screen</i>	108
5.1.17	Halaman Tampil <i>Section</i>	110
5.1.18	Halaman Tambah Data <i>Section</i>	112
5.1.19	Halaman <i>Edit</i> Data <i>Section</i>	114
5.1.20	Halaman Hapus Data <i>Section</i>	119
5.1.21	Halaman <i>Corona Updates</i>	121
5.2	Pengujian Fungsionalitas Perangkat Lunak	123
5.3	Hasil Pengujian Terhadap Pengguna	143
BAB VI. PENUTUP		144
6.1	Kesimpulan	144
6.2	Saran.....	144
DAFTAR PUSTAKA		145

DAFTAR GAMBAR

Gambar 3.1 Arsitektur <i>Microservice</i> [8].....	11
Gambar 4.1 Analisis Sistem CMS	18
Gambar 4. 2 Use Case Diagram CMS (Angular).....	21
Gambar 4. 3 Use Case Diagram CMS Tipe (Next.js).....	25
Gambar 4. 4 Use Case Diagram CMS Komponen (Next.js).....	27
Gambar 4. 5 Use Case Diagram CMS <i>Screen</i> (Next.js)	29
Gambar 4. 6 Use Case Diagram CMS <i>Section</i> (Next.js).....	31
Gambar 4. 7 Use Case Diagram <i>Website Corona Updates</i>	33
Gambar 4.8 Overview Sistem Arsitektur CMS Pada Angular.....	37
Gambar 4. 9 Overview Arsitektur Sistem CMS Pada Next.js	38
Gambar 4. 10 Overview Arsitektur Sistem Pada <i>Website Corona Updates</i>	39
Gambar 4. 11 Arsitektur Perangkat lunak CMS <i>Framework Angular</i>	40
Gambar 4. 12 Arsitektur Perangkat lunak CMS <i>Framework Next js</i>	41
Gambar 4. 13 Arsitektur Perangkat lunak CMS <i>Framework Next js</i>	42
Gambar 4. 14 Class Diagram Pada Next.js	43
Gambar 4.15 Halaman Tambah Data CMS <i>framework Angular</i>	51
Gambar 4.16 Halaman <i>Edit</i> Data CMS <i>framework Angular</i>	52
Gambar 4.17 Halaman Hapus Data CMS <i>framework Angular</i>	53
Gambar 4.18 Halaman Login Pada CMS <i>Framework Next.js</i>	54
Gambar 4.19 Halaman Menampilkan data Tipe	55
Gambar 4.20 Halaman untuk mencari data tipe.....	55
Gambar 4.21 Halaman Login Pada CMS <i>Framework Next.js</i>	56
Gambar 4.22 Halaman <i>Edit</i> Data Tipe	57
Gambar 4.23 Halaman Hapus Data Tipe	58
Gambar 4.24 Halaman Tampil Data Komponen.....	59
Gambar 4.25 Halaman Cari Data Komponen	59
Gambar 4.26 Halaman Tambah Data Komponen	60
Gambar 4.27 Halaman <i>Edit</i> Data Komponen	61

Gambar 4.28 Halaman Hapus Data Komponen	62
Gambar 4.29 Halaman Tampil Data <i>Screen</i>	63
Gambar 4.30 Halaman Cari Data Komponen	63
Gambar 4.31 Halaman Tambah Data <i>Screen</i>	64
Gambar 4.32 Halaman <i>Edit Data Screen</i>	65
Gambar 4.33 Halaman Hapus Data <i>Screen</i>	66
Gambar 4.34 Halaman Tampil <i>Section</i>	67
Gambar 4. 35 Halaman Tambah Data <i>Section</i>	68
Gambar 4. 36 Halaman <i>Edit Data Section</i>	69
Gambar 4.37 Halaman Hapus Data <i>Section</i>	70
Gambar 4.38 Halaman Corona <i>Updates</i>	71
Gambar 5.1 Antarmuka Halaman Tambah Data CMS <i>Framework</i> Angular.....	72
Gambar 5.2 Fungsi Mengolah Data Inputan Dari <i>User</i> Pada Angular	73
Gambar 5.3 HTTP Client Pada Angular Untuk Tambah Data CMS	73
Gambar 5.4 Logic pada sisi server dengan Node.js dan Express.js	74
Gambar 5. 5 Model pada sisi server.....	74
Gambar 5.6 Antarmuka Pada <i>Edit Data</i> CMS Menggunakan <i>Framework</i> Angular	75
Gambar 5.7 Fungsi <i>Edit Data</i> Pada Angular	76
Gambar 5.8 HTTP Client <i>Update</i> Pada Angular Untuk <i>Edit Data</i> CMS.....	76
Gambar 5.9 Logic <i>Update</i> Pada Sisi Server Dengan Node.js dan Express.js.....	77
Gambar 5.10 Model <i>Update</i> Pada Sisi Server	77
Gambar 5.11 Antarmuka Hapus Data CMS Menggunakan Framewok Angular ..	78
Gambar 5.12 Fungsi Hapus Pada CMS menggunakan <i>framework</i> Angular.....	79
Gambar 5.13 HTTP Client Hapus pada Angular	79
Gambar 5.14 Logic <i>Delete</i> Pada Sisi Server Dengan Node.js dan Express.js	80
Gambar 5.15 Model <i>Update</i> Pada Sisi Server	80
Gambar 5.16 Antarmuka Login Untuk CMS Menggunakan <i>Framework</i> Next.js.	81
Gambar 5.17 Setting Untuk Akses Firebase	82
Gambar 5.18 Fungsi <i>User Service</i> Login Pada Next.js.....	82
Gambar 5.19 Antarmuka Halaman Tampil Tipe.....	83

Gambar 5.20 Antarmuka Halaman Cari Tipe	83
Gambar 5.21 Fungsi <i>getdata</i> pada tipe	84
Gambar 5.22 Fungsi <i>handleChange</i> pada search tipe	84
Gambar 5.23 Fungsi <i>GetDataTypes Service Layout</i> Pada Tipe	85
Gambar 5.24 Antarmuka Halaman Tambah Data Tipe	85
Gambar 5.25 Fungsi Tambah Data Pada Tipe	86
Gambar 5.26 <i>Service Layout</i> Tambah Data Tipe	86
Gambar 5. 27 Antarmuka Halaman <i>Edit</i> Data Tipe	87
Gambar 5.28 Fungsi <i>Edit</i> Data Pada Tipe	88
Gambar 5.29 <i>Service Layout Edit</i> Data Tipe	88
Gambar 5. 30 Antarmuka Halaman Hapus Data Tipe	89
Gambar 5.31 Fungsi Hapus Data Pada Tipe	90
Gambar 5.32 <i>Service Layout</i> Hapus Data Tipe	90
Gambar 5.33 Antarmuka Halaman Tampil Komponen	91
Gambar 5.34 Antarmuka Halaman Tampilan Cari Komponen	92
Gambar 5.35 Fungsi <i>getdata</i> Pada Komponen	92
Gambar 5.36 <i>Service Layout</i> Hapus Data Komponen	93
Gambar 5.37 Antarmuka Halaman Tambah Data Komponen	93
Gambar 5.38 Fungsi Tambah Data Pada Komponen	94
Gambar 5.39 <i>Service Layout</i> Tambah Data Tipe	95
Gambar 5.40 Antarmuka Halaman <i>Edit</i> Data Komponen	96
Gambar 5.41 Fungsi <i>Edit</i> Data Pada Komponen	97
Gambar 5.42 <i>Service Layout Edit</i> Data Komponen	98
Gambar 5. 43 Antarmuka Halaman Hapus Data Komponen	99
Gambar 5.44 Fungsi Hapus Data Pada Komponen	100
Gambar 5. 45 <i>Service Layout</i> Hapus Data Tipe	100
Gambar 5.46 Antarmuka Halaman Tampil <i>Screen</i>	101
Gambar 5. 47 Antarmuka Halaman Cari <i>Screen</i>	101
Gambar 5. 48 Fungsi <i>getdata</i> pada <i>screen</i>	102
Gambar 5. 49 Fungsi <i>handleChange</i> pada search <i>screen</i>	102
Gambar 5. 50 Fungsi <i>GetType Service Layout</i>	103

Gambar 5.51 Antarmuka Halaman Tambah Data <i>Screen</i>	103
Gambar 5.52 Fungsi Tambah Data Pada <i>Screen</i>	104
Gambar 5.53 <i>Service Layout</i> Tambah Data <i>Screen</i>	104
Gambar 5.54 Antarmuka Halaman <i>Edit Data Screen</i>	105
Gambar 5.55 Fungsi <i>Edit Data</i> Pada <i>Screen</i>	106
Gambar 5.56 <i>Service Layout Edit Data Screen</i>	107
Gambar 5.57 Antarmuka Halaman Hapus Data <i>Screen</i>	108
Gambar 5.58 Fungsi Hapus Data Pada <i>Screen</i>	109
Gambar 5.59 <i>Service Layout</i> Hapus Data <i>Screen</i>	109
Gambar 5.60 Antarmuka Halaman Tampil <i>Section</i>	110
Gambar 5.61 Fungsi <i>getdata</i> Pada <i>Section</i>	111
Gambar 5. 62 <i>Service Layout</i> Pada Tampil Data <i>Section</i>	111
Gambar 5.63 Antarmuka Halaman Tambah Data <i>Screen</i>	112
Gambar 5.64 Fungsi Tambah Data Pada <i>Section</i> (1).....	112
Gambar 5.65 Fungsi Tambah Data Pada <i>Section</i> (2).....	113
Gambar 5.66 <i>Service Layout</i> Tambah Data <i>Section</i>	113
Gambar 5.67 Antarmuka Halaman <i>Edit Data Section</i>	114
Gambar 5.68 Fungsi <i>Edit Data</i> Pada <i>Section</i> (1).....	115
Gambar 5. 69 Fungsi <i>Edit Data</i> Pada <i>Section</i> (2).....	115
Gambar 5.70 <i>Service Layout Edit Data Section</i>	116
Gambar 5.71 Fungsi <i>Edit Position Data Section</i> (1)	117
Gambar 5. 72 Fungsi <i>Edit Position Data Section</i> (2)	117
Gambar 5.73 Antarmuka Halaman Hapus Data <i>Section</i>	119
Gambar 5.74 Fungsi Hapus Data Pada <i>Section</i>	120
Gambar 5.75 <i>Service Layout Delete Data Section</i>	120
Gambar 5. 76 Antarmuka Halaman Corona <i>Update</i>	121
Gambar 5.77 Fungsi <i>getdata</i> Jurnal dan Artikel Pada Halaman Corona <i>Updates</i>	122

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian.....	8
Tabel 4. 1 Use Case Diagram Mengelola Data	21
Tabel 4. 2 Use Case Diagram Mengelola Data CMS Tipe.	25
Tabel 4. 3 Use Case Diagram CMS Komponen	27
Tabel 4. 4 Use Case Diagram CMS <i>Screen</i>	29
Tabel 4. 5. Use Case Diagram CMS <i>Section</i> (Next.js)	31
Tabel 4. 6 Use Case Diagram <i>Website Corona Updates</i>	33
Tabel 4.7 Struktur Tabel Kelas <i>MasterType</i>	44
Tabel 4.8 Fungsi <i>GetAllType</i> Pada Data <i>Master Tipe</i>	44
Tabel 4.9 Fungsi <i>InsertType</i> pada Data <i>Master Tipe</i>	44
Tabel 4.10 Fungsi <i>UpdateType</i> Pada Data <i>Master Tipe</i>	44
Tabel 4.11 Fungsi <i>deleteType</i> Pada Data <i>Master Tipe</i>	45
Tabel 4.12 Struktur Tabel Kelas <i>MasterComponent</i>	45
Tabel 4.13 Fungsi <i>GetAllComponent</i> Pada Data <i>Master Component</i>	46
Tabel 4.14 Fungsi <i>insertComponent</i> Pada Data <i>Master Component</i>	46
Tabel 4.15 Fungsi <i>UpdateComponent</i> Pada Data <i>Master Component</i>	46
Tabel 4.16 Fungsi <i>DeleteComponent</i> Pada Data <i>Master Component</i>	47
Tabel 4. 17 Struktur Tabel Kelas <i>MasterSections</i>	47
Tabel 4. 18 Fungsi <i>getAllSections</i> Pada Data <i>Master Sections</i>	48
Tabel 4. 19 Fungsi <i>insertSections</i> Pada Data <i>Master Sections</i>	48
Tabel 4. 20 Fungsi <i>updateSections</i> Pada Data <i>Master Sections</i>	48
Tabel 4. 21 Fungsi <i>deleteSections</i> Pada Data <i>Master Sections</i>	49
Tabel 4. 22 Struktur Tabel Kelas <i>MasterScreens</i>	49
Tabel 4. 23 Fungsi <i>getAllScreen</i> Pada Data <i>Master Screens</i>	49
Tabel 4. 24 Fungsi <i>insertScreens</i> Pada Data <i>Master Screens</i>	50
Tabel 4. 25 Fungsi <i>getAllScreens</i> Pada Data <i>Master Screens</i>	50
Tabel 4. 26 Fungsi <i>getAllScreens</i> Pada Data <i>Master Screens</i>	50

Tabel 5. 1 Pengujian Fungsionalitas Perangkat Lunak123
Tabel 5. 2 Hasil Pengujian Terhadap Pengguna143



INTISARI

PENGEMBANGAN DAN *MAINTENANCE* APLIKASI KESEHATAN PADA PT. GLOBAL URBAN ESENSIAL

Hendrikus Adi Purnama
160709007

PT. Global Urban Esensial merupakan perusahaan startup kesehatan yang merupakan bagian dari Deka Group yang bergerak di bidang kesehatan yang sudah berdiri sejak tahun 2015 hingga sekarang. Saat ini perusahaan Global Urban Esensial telah menghasilkan beberapa produk seperti GoApotik, Teman Diabetes, GueSehat, Teman Bumil, Ifocus, dan D2D (*Doctor to Doctor*). Aplikasi yang dihasilkan sudah banyak membantu masyarakat terutama dalam bidang kesehatan.

Penelitian ini mendiskusikan perubahan arsitektur pada aplikasi atau *revamp* dari *monolithic* ke *microservice* agar *service* yang dihasilkan lebih terstruktur dan mudah untuk dikembangkan. Sistem informasi yang dibangun menjadi dapat terbagi dan menyediakan layanan secara detail. Dalam hal ini beberapa *service* yang besar dipecah dalam *service* kecil agar dapat digunakan untuk membuat CMS (*Content Management Service*) serta pengembangan *website* corona pada aplikasi D2D dengan menggunakan JavaScript.

Tujuan adanya *maintenance* dan pengembangan aplikasi pada produk dari studi kasus di PT. Global Urban Esensial yang selanjutnya disebut PT.GUE memberikan kenyamanan bagi pengguna dengan sistem yang ringan, *high performance*, dan lebih mudah untuk dikembangkan terutama dalam pengembangan CMS yang membuat tampilan pada aplikasi GueSehat lebih dinamis, sedangkan pada *website* Corona *Updates* memberikan fitur yang menampilkan jurnal dan artikel tentang COVID-19 dengan tidak memakan *resource* yang besar dan waktu yang lama.

Kata Kunci: *Revamp*, *Microservice*, JavaScript, Express.js, Node.js

Dosen Pembimbing I : Dr. Ir. Alb. Joko Santoso, M.T.

Dosen Pembimbing II : Dr. Andi Wahyu Rahardjo Emanuel, BSEE., MSSE.

Jadwal Sidang Tugas Akhir : Rabu, 08 Juli 2020

BAB I. PENDAHULUAN

1.1 Latar Belakang

Perusahaan GUE (PT. Global Urban Esensial) merupakan bagian dari Dexa Group yang bergerak dibidang kesehatan. PT. GUE sendiri berbeda dengan beberapa anak perusahaan Dexa yang lain, dikarenakan PT. GUE berfokus pada bagian teknologi dan informasi untuk informasi dibidang kesehatan. Banyak produk yang telah dihasilkan oleh PT. GUE seperti GoApotik, Teman Diabetes, GueSehat, Teman Bumil, iFocus, dan D2D (*Doctor to Doctor*). Pada aplikasi GueSehat merupakan produk yang memberikan edukasi kesehatan kepada pengguna seperti artikel, direktori, info kesehatan, dan forum, sedangkan aplikasi Doctor to Doctor (D2D) merupakan salah satu aplikasi yang dikembangkan oleh PT. GUE sebagai aplikasi yang digunakan oleh para dokter sebagai sarana untuk berbagi literasi di dunia kesehatan.

Sebagai salah satu aplikasi yang populer di PT. GUE, GueSehat diliris pada tanggal 9 agustus 2017 dengan 100 ribu lebih orang yang telah mengunduh aplikasi tersebut pada Google Play Store, demikian juga pada App Store aplikasi ini dirilis tanggal 1 September 2017. Aplikasi ini dibangun dengan tujuan memberikan edukasi dan informasi kepada *user* GueSehat untuk memberikan manfaat dan *tools* yang dibutuhkan setiap orang untuk menjadi sehat. Terdapat beberapa fitur didalamnya seperti forum yang digunakan untuk berbagi pengalaman di bidang kesehatan, Direktori yang digunakan *user* untuk mencari informasi dokter yang kini terdapat lebih dari 4.500 dokter dengan berbagai macam spesialis dan beberapa fitur artikel dan tips lainnya secara lengkap dan akurat. Aplikasi GueSehat mengadopsi arsitektur *monolithic* dimana satu *server* menangani semua layanan, dengan banyaknya konten pada aplikasi GueSehat maka akan sangat sulit dalam mengatur *content* yang akan ditampilkan maka perlu dibuatnya CMS (*Content Management System*) untuk mengatur konten pada halaman di aplikasi GueSehat.

Sedangkan aplikasi D2D yang merupakan aplikasi untuk dokter yang

terdapat berbagai macam fitur unggulan seperti jurnal ilmiah, akses video, *list event*, serta fitur *guideline* yang berguna sebagai pedoman. Sehubungan dengan bertambahnya kebutuhan pada aplikasi D2D terkait dengan isu kesehatan yang sedang dihadapi dunia yaitu COVID-19 (*Corona Virus Disease 2019*) maka dilakukan penambahan fitur pada *website* D2D yaitu fitur *Corona Updates* yang memuat kumpulan jurnal dan artikel mengenai COVID-19 dengan mengimplementasikan Next.js pada *website* tersebut.

1.2 Rumusan Masalah

Berdasarkan dari latar belakang, permasalahan yang muncul diantaranya yaitu:

1. Bagaimana menerapkan arsitektur *microservice* pada aplikasi GueSehat agar diimplementasikan pada CMS (*Content Management System*) untuk mengelola layout konten pada aplikasi GueSehat?
2. Bagaimana mengimplementasikan *framework* Next.js untuk *website* D2D pada fitur *Corona Updates* untuk menampilkan jurnal-jurnal terkait dengan COVID-19?

1.3 Batasan Masalah

Penelitian ini memiliki batasan masalah sebagai berikut:

1. Aplikasi yang di-*maintenance* dan dikembangkan adalah aplikasi GueSehat pada fitur CMS (*Content Management System*) dan *website* D2D pada fitur *Corona Updates*.
2. Aplikasi CMS (*Content Management System*) yang di-*maintenance* hanya dapat digunakan oleh tim produk pada perusahaan tersebut
3. Pembahasan akan terbatas oleh beberapa informasi yang tidak boleh dipublikasikan oleh pihak PT.Global Urban Esensial.
4. Dalam *maintenance* GueSehat dan pengembangan aplikasi D2D menggunakan arsitektur *microservice* dengan menggunakan 2 *framework* yaitu Angular dan Next.js.

1.4 Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut.

1. Membuat CMS (*Content Management System*) untuk GueSehat dengan menggunakan arsitektur *microservice* agar mudah dalam mengatur konten dan informasi pada *dashboard*.
2. Membuat *Corona Site* untuk *platform* D2D berbasis *website* dengan menggunakan *framework* Next.js.

1.5 Metode Penelitian

a. Studi Literatur

Metode studi literatur digunakan dengan informasi yang berkaitan dengan *microservice* dan penggunaan *stack* seperti Node.js, Angular, dan Next.js.

b. Analisis

Analisis digunakan untuk mengetahui apa yang dibutuhkan oleh aplikasi serta menentukan *best practice* yang harus digunakan agar tidak mengurangi *performance*.

c. Implementasi Perangkat Lunak

Untuk tahapan implementasi ini, akan dilakukan pembuatan fungsi atau *code* sesuai dengan *requirement*. Pembuatan program dilakukan dengan menggunakan bahasa pemrograman yang sesuai dengan aturan – aturan dalam bahasa pemrograman yang digunakan. Hasil dari metode ini berupa CMS (*Content Management System*) pada GueSehat dan fitur *corona updates* berbasis *website* yang menggunakan bahasa pemrograman Javascript dan *framework* Node.js , Angular, dan Next.js yang digunakan sebagai *stack* perusahaan.

d. Pengujian Perangkat Lunak

Sesudah perancangan aplikasi selesai, dilakukan sesi pengujian yang bertujuan untuk melihat apakah aplikasi telah cocok dengan spesifikasi yang telah ditetapkan.

e. Pelaporan

Pada metode ini dilakukan untuk pembuatan laporan tugas akhir dimana

tujuan dari pelaporan ini merupakan untuk mengenali perkembangan ataupun penjelasan dalam pembuatan program yang nantinya pelaporan ini bisa digunakan untuk bahan penilaian dalam pembuatan program berikutnya.

1.6. Sistematika Penulisan

Sistematika penulisan untuk penelitian ini adalah:

BAB I: Pendahuluan

Pada bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, alat dan bahan, metodologi penelitian, dan sistematika penulisan laporan yang terkait dengan proses penelitian ini.

BAB II: Tinjauan Pustaka

Bab ini berisi tentang penjelasan singkat dan ringkasan mengenai penelitian terdahulu yang berhubungan atau memiliki kesamaan dengan permasalahan yang akan dibahas oleh penulis di dalam tugas akhir ini.

BAB III: Landasan Teori

Pada bab ini berisi dasar-dasar teori yang melandasi serta mendukung dalam implementasi aplikasi beserta proses yang berhubungan dengan pembangunan aplikasi.

BAB IV: Analisis dan Perancangan Sistem

Bab ini berisi penjelasan serta analisis perancangan sistem dan implementasi sistem yang telah dibuat.

BAB V: Implementasi dan Pengujian Sistem

Bab ini berisi tentang penjelasan dari implementasi serta hasil dari pengujian sistem.

BAB VI: Penutup

Bab ini berisi kesimpulan mengenai *project* yang telah dibuat beserta saran-saran yang berguna bagi pengembangan lebih lanjut.

BAB II. TINJAUAN PUSTAKA

Pada penelitian pengembangan dan *maintenance* aplikasi kesehatan pada PT. Global Urban Esensial, langkah awal yang dilakukan ialah menggali sumber informasi yang mengacu pada berbagai penelitian yang telah dibuat sebelumnya baik dengan menggunakan buku atau jurnal yang telah diterbitkan untuk umum. Proses penggalian literasi dilakukan dengan tujuan untuk mendapat informasi yang memiliki hubungan dengan judul terkait, *microservice*, dan penggunaan Node.js sebagai *server*. Untuk informasi yang didapat juga berguna sebagai landasan teori ataupun kajian.

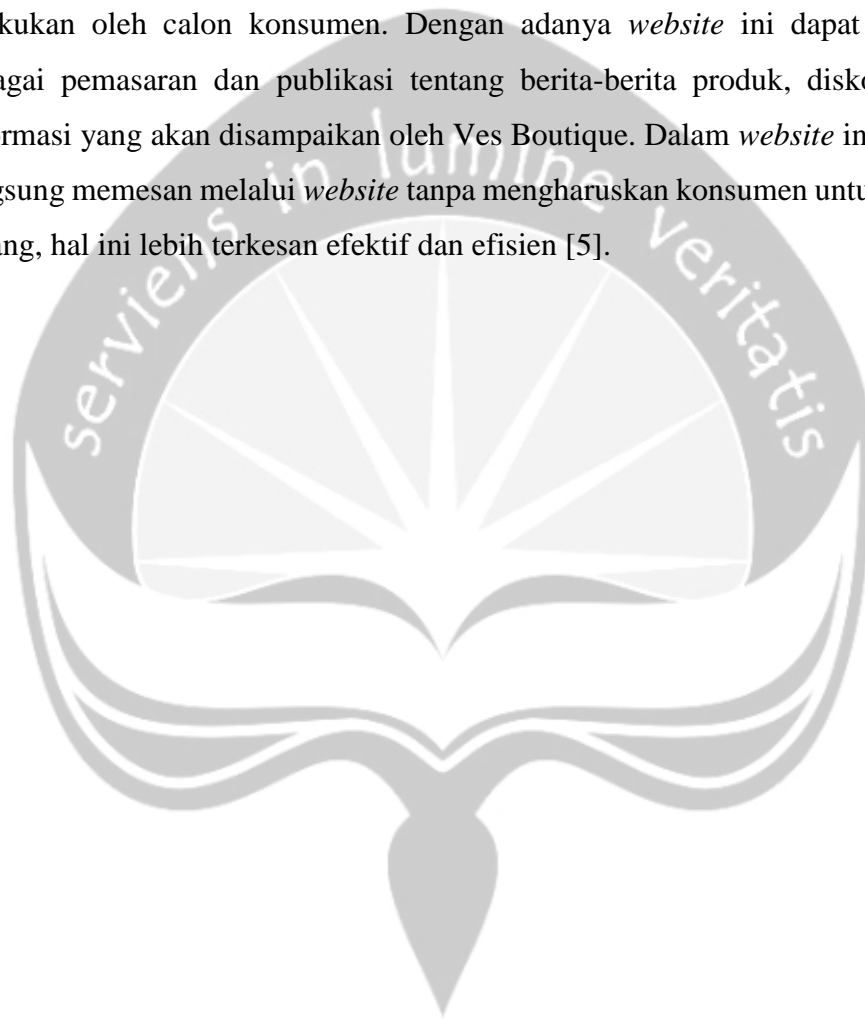
Terdapat penelitian dan perancangan aplikasi yang dilakukan oleh Iqbal C.R., dkk dari Institut Teknologi Sepuluh Nopember (ITS) tentang implementasi *client* SIP berbasis web dengan menggunakan HTML5 dan Node.js. Penulisan tugas akhir ini membahas tentang penggunaan *microservice* untuk mengimplementasikan *client* SIP. Permasalahan yang dialami adalah dari hasil implementasi yang dilakukan, dalam penggunaan HTML5 adalah kompatibilitas dimana fitur WebRTC belum diterapkan pada beranda yang biasa digunakan dan jika dilakukan dengan *monolithic* maka aplikasi akan sulit untuk dikembangkan. Berlandaskan kasus yang terjadi, hingga dari itu penulis membuat sebuah *microservice* yang dapat memberikan kompatibilitas yang baik kepada pengguna [1].

Di dalam penelitian yang dilakukan oleh Siambaton dan Fakhiza dari Universitas Sumatera Utara tentang aplikasi CMS (*Content Management System*). Penelitian ini membahas tentang aplikasi CMS yang dapat mempermudah dalam pengelolaan *website*. Dengan menampilkan beberapa konten, *user*, dan kategori. Dengan menggunakan aplikasi Joomla di dalam *web service* tersebut dapat digunakan dengan berbagai fitur khususnya dapat digunakan oleh aplikasi lain fitur tersebut meliputi pengelolaan *content*, pengelolaan *section* dan *category*, pengelolaan *user*, pengelolaan media serta mengelola *user* pada *web service* tersebut. Dengan jenis fitur tersebut hasil pengujian menunjukkan aplikasi tersebut bersifat dinamis sehingga lebih mudah mengelola konten dengan baik [2].

Di dalam penelitian yang dilakukan oleh Ulinuha dan Irawan dari Universitas Muria Kudus tentang CMS (*Content Management System*) Untuk Tes *Online* Mahasiswa Baru Pada Universitas Muria Kudus. Pada penelitian ini dilatarbelakangi oleh perkembangan teknologi yang semakin masif, dengan akses internet yang sangat mudah menjadi tolak ukur bahwa masyarakat telah siap berinteraksi dengan *website*, oleh karena itu Universitas Muria Kudus menerapkan CMS untuk melakukan tes *online* bagi calon mahasiswa baru. Hasil dari penelitian ini sendiri adalah sebuah CMS yang berupa tes *online*. Dalam CMS ini dibagi beberapa modul diantaranya untuk mengatur pendaftaran, mengatur web, manajemen pengaturan, serta mengatur *user*. Sistem ini dikembangkan dalam bahasa pemrograman PHP dengan basis data MySQL sehingga dapat dengan semua *browser*. Hasilnya nanti CMS tes *online* ini bukan hanya digunakan oleh satu universitas. Tetapi dapat dipergunakan oleh universitas lain dengan *setting* tertentu [3].

Dalam penelitian yang dilakukan oleh Suryotrisogko dari Institut Teknologi Sepuluh Nopember tentang Arsitektur *Microservice* untuk resiliensi sistem informasi. Pada penelitian ini *design pattern* arsitektur dapat dikembangkan dengan aspek kualitas dari resiliensi yang berarti mampu bertahan terhadap sesuatu kerusakan ataupun perubahan didalam sebuah infrastruktur *microservice*. *Microservice* sendiri berjalan pada sisi *server side* atau *backend* yang akan meliputi Rest API yang terproteksi dengan JWT *autentifikasi*. Pada dasarnya pola komunikasi antara *frontend* (sisi *client*) dan *back-end* (sisi *server*) berjalan secara *asynchronous*, sehingga *server* dapat mengingat token *session* sebagaimana seperti pola komunikasi *client* dan *server*. Dengan pendekatan *Microservice – Docker* yang telah dikembangkan dengan *Spring Boot* pola yang dihasilkan oleh Rest *Service* atau Rest API dengan format JSON memberikan implementasi kemudahan yang tidak akan membebani *server*. Dalam sudut *end user* bila aspek resiliensi ini diterapkan maka dapat diamati pada stabilitas sistem karena tidak secara langsung gangguan ataupun kerusakan dapat dirasakan, dapat disimpulkan juga bahwa aspek resiliensi dapat diandalkan pada arsitektur *microservice* ini [4].

Di dalam penelitian yang dilakukan oleh Hasugian dari STMIK Pelita Nusantara tentang Perancangan *Website* Sebagai Media Promosi dan Informasi, penulis merancang *website* yang dapat digunakan sebagai media promosi dan informasi pada Ves Boutique, beberapa komponen di dalamnya seperti halaman admin yang diperutukan untuk mengontrol data pada *website*. Terdapat halaman admin yang terdapat unit kategori, produk, *order*, *login* serta *slide*. Sedangkan untuk komponen halaman *user*, digunakan untuk pemesanan dan mengelola kontak yang dilakukan oleh calon konsumen. Dengan adanya *website* ini dapat digunakan sebagai pemasaran dan publikasi tentang berita-berita produk, diskon ataupun informasi yang akan disampaikan oleh Ves Boutique. Dalam *website* ini pun dapat langsung memesan melalui *website* tanpa mengharuskan konsumen untuk langsung datang, hal ini lebih terkesan efektif dan efisien [5].



Tabel 2.1 Perbandingan Penelitian

No.	Judul	Peneliti	Tujuan Penelitian	Platform	Kesimpulan
1	Implementasi Klien SIP Berbasis Web Dengan Menggunakan HTML5 dan NodeJS	Iqbal C.R., dkk. [1]	Tujuan penelitian ini adalah menggunakan <i>microservice</i> untuk handle layanan pada SIP	Web	Kesimpulan yang bisa ditarik dari implementasi client SIP berbasis <i>website</i> ini merupakan HTML5 serta Node.js mampu diimplementasikan sebagai aplikasi <i>website</i> client SIP serta video chat. Komunikasi audio serta video yang memakai <i>codec</i>
2	Aplikasi <i>Content Management System</i> (CMS) Pada Joomla Untuk Membuat Web <i>Service</i>	Siambaton dan Fakhriza [2]	Membangun aplikasi CMS untuk mempermudah pengolahan content dan manajemen informasi pada <i>website</i>	Aplikasi Joomla	Kesimpulan yang dapat ditarik adalah perangkat lunak memiliki fitur dasar dalam penyediaan <i>webservice</i> serta dapat mengelola konten sehingga bersifat dinamis dan mudah untuk mengelola informasi di dalamnya.

3	<i>Content Management System (CMS) Untuk Tes Online Mahasiswa Baru Pada Universitas Muria Kudus.</i>	Ulinuha dan Irawan [3]	Membangun aplikasi <i>Content Management System (CMS) Untuk Tes Online Mahasiswa Baru Pada Universitas Muria Kudus</i>	PHP dan MySQL	Aplikasi mampu digunakan sebagai seleksi tes online yang telah dibangun dengan PHP dan MySQL. Serta nantinya CMS tes online ini dapat digunakan oleh instansi atau universitas lain.
4	Arsitektur <i>Microservice</i> untuk Resiliensi Sistem Informasi.	Suryotrisogko [4]	Membangun <i>design pattern</i> arsitektur <i>microservice</i> yang mampu beradaptasi terhadap masalah atau kesalahan pada infrastruktur	Docker dan <i>Web Service</i>	Aplikasi dengan arsitektur <i>microservice</i> lebih tahan terhadap masalah tanpa berikatan dengan <i>service</i> lain sehingga memberi kemudahan dalam melakukan evaluasi dan memperbaiki ketika terdapat masalah pada <i>service</i>
5	Perancangan <i>Website</i> Sebagai Media Promosi dan Infromasi	Hasugian [5]	Mengembangkan <i>website</i> berbasis media informasi yang digunakan untuk promosi dan bertransaksi	PHP dan MySQL	Pada perancangan <i>website</i> ini mempermudah dalam memasarkan serta mempublikasikan informasi serta penyampaian yang

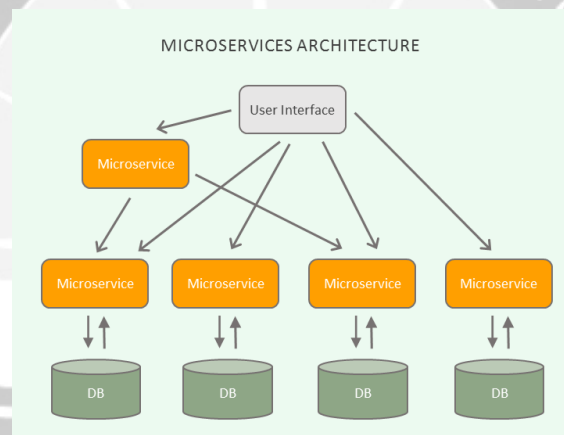
					dilakukan oleh Ves Boutique.
6	Pengembangan dan <i>Maintenance</i> Aplikasi Kesehatan Pada PT. Global Urban Esensial	*) Purnama	Mengembangkan <i>website</i> serta melakukan <i>maintenance</i> terhadap <i>service</i> dan membuat <i>Content Management System</i> pada aplikasi GueSehat serta Corona <i>website</i> pada aplikasi D2D.	Node.js	Pada pengembangan <i>website</i> ini akan memudahkan dalam mengatur <i>management content</i> yang ada pada aplikasi GueSehat serta mengembangkan <i>website</i> Corona yang digunakan untuk memudahkan dalam melihat list jurnal.

*) Sedang melakukan penelitian.

BAB III. LANDASAN TEORI

3.1. *Microservice*

Layanan *microservice* adalah metode populer untuk merancang aplikasi berbasis *cloud* yang dapat dipecah menjadi beberapa bagian fungsional. Aplikasi berbasis *microservice* bergantung pada *message* yang lewat untuk komunikasi dan memisahkan setiap layanan *microservice*, dengan penggunaan *microservice* sendiri dapat melayani berbagai macam *service* dengan tidak mengganggu *service* lainnya. Aplikasi dapat berisi ratusan layanan mikro, hal ini memudahkan untuk mempertimbangkan tentang aspek pengembangan. Secara khusus untuk pengembangan yang menentukan *performance* pada layanan tersebut [6].



Gambar 3.1 Arsitektur *Microservice* [7].

Microservice diimplementasikan dengan *pattern* arsitektur yang secara drastis mempengaruhi hubungan antara beberapa *logic* dan *database*. Dengan tidak menggabungkan beberapa *logic* dan *database*, masing-masing *services* akan memiliki *logic* dan skema *database* tersendiri. Namun dengan memecah *service* secara kecil dan terfokus akan sangat penting pada penerapan arsitektur *microservice*. Masing-masing *service* akan memiliki *logic* tersendiri serta sebuah *service* tidak mempengaruhi *service* lainnya. Sebagai contoh jika salah satu *service* pada aplikasi tersebut mengalami gangguan maka *service* lainnya tidak akan terpengaruh, hal ini berbeda dengan *monolithic* yang akan

mengganggu semua *service* dikarenakan ketergantungan antar layanan. Selain itu, *services* dapat menggunakan jenis *database* dan bahasa pemrograman yang paling sesuai dengan kebutuhannya [8] .

3.2. Content Management System

CMS (*Content Management System*) merupakan salah satu sistem yang dipakai untuk mengelola, memperbaharui, serta menambahkan konten yang ada di aplikasi *web* ataupun *mobile*. Sebuah konten biasanya memacu pada teks, gambar, maupun bentuk dari tampilan. CMS sangat memberikan kemudahan kepada pengguna tanpa harus tau hal-hal yang bersifat teknis untuk mengubah tampilan yang dinamis.

CMS sangat memberikan banyak keuntungan terutama dalam mengatur posisi-posisi yang ditampilkan pada *website* dengan berbagai jenis *file* mulai dari foto, *video*, teks, maupun *audio*. CMS saat ini banyak sekali digunakan dengan berbasis *website*. Dengan kelebihan agar dapat dengan mudah diakses dimanapun hanya dengan menggunakan akses internet dan *tool* seperti *browser engines* [9].

3.3. JavaScript

JavaScript merupakan bahasa scripting yang sering dipakai sebagai pengembangan *website*. Saat ini JavaScript memiliki kelebihan yaitu tidak memerlukan compiler untuk menjalankannya seperti pada bahasa C dan C++. Kode JavaScript biasanya berjalan langsung di browser web dan ditulis dengan menggunakan *scripting* yang ringan dan cepat. Javascript dapat digunakan sebagai menyempurnakan tampilan pada sisi *client* agar *website* menjadi interaktif dan *user friendly*. Untuk penggunaan JavaScript akan disisipkan pada *element head* pada HTML [10].

Tidak hanya digunakan pada *client-side* saja, saat ini JavaScript banyak digunakan untuk meng-*handle* pada sisi *server* atau *server-side* dengan pengembangan aplikasi pada Node.js yang digunakan sebagai *run time*. Berbeda dengan JavaScript yang berjalan di sisi *client*, Node.js dapat dijalankan dengan

dukungan V8 Engine buatan Google dan *dependency* atau modul bawaan yang terintegrasi dengan HTTP, *security*, *filesystem*, dan modul penting lainnya[11].

3.4. Node.js

Node.js (Node) adalah *platform* pengembangan *source* terbuka untuk mengeksekusi sisi *server* kode pada JavaScript. Node berguna untuk mengembangkan aplikasi yang memerlukan koneksi terus-menerus dari browser ke *server* dan sering digunakan untuk aplikasi waktu nyata seperti *chat*, *news feeds* and *web push notifications*.

Node.js dimaksudkan untuk dijalankan pada *server* HTTP khusus dan untuk menggunakan *single thread* dengan satu proses pada satu waktu. Aplikasi Node.js *event-based* dan dijalankan secara tidak sinkron. Kode yang dibangun pada *platform* Node tidak mengikuti model tradisional seperti mengirim, memproses, dan menerima. Node.js memproses *request* dengan cara *asynchronous* yaitu setiap *request* yang masuk akan berada di dalam tumpukan *event* yang konstan dan akan mengirimkan *request* kecil satu demi satu tanpa menunggu *response*. Ini adalah pergeseran dari model utama atau biasanya disebut *synchronous* yang menjalankan proses yang lebih besar, lebih kompleks, dan menjalankan beberapa perintah secara bersamaan, dengan masing-masing perintah akan menunggu *response* yang sesuai sebelum melanjutkan *request* lainnya.

Salah satu keuntungan utama Node.js, menurut penciptanya Ryan Dahl, adalah bahwa ia tidak memblokir *input / output (I / O)*. Jadi dalam suatu *request* yang besar dapat dijalankan sekaligus tanpa menunggu adanya *response* antara satu fungsi dengan fungsi lainnya sehingga hal ini dapat mempercepat proses pengolahan data dan langsung mengembalikan data jika proses pada *request* sudah terpenuhi [12].

3.5. Express.js

Express.js adalah sebuah *framework* atau kerangka kerja yang terdapat dalam Node.js yang mudah dikembangkan untuk pengembangan aplikasi web,

service API, routing, maupun *security*. Dikembangkannya Express.js akan berguna pada penggunaan *design pattern* yang dapat disesuaikan dengan arsitektur apapun sehingga sangat *powerfull* dan fleksible. Dengan pembuatan arsitektur degan Express.js, API yang digunakan juga sangat ringan dan tidak memakan *resource* yang sangat banyak sehingga sangat memangkas *cost* yang digunakan untuk pengembangan *website* selanjutnya [11].

Dalam hal ini dapat mengatur middleware pada permintaan HTTP serta dapat membuat sebuah routing yang dapat digunakan dan mempunyai fungsi yang berbeda berdasarkan dengan metode dan HTTP atau URL yang digunakan. Dan juga membuat secara dinamis dalam pages HTML berdasarkan template. Express.js juga sangat dinamis dan lebih *powerfull* karena ada banyak modul yang tersedia di dalam NPM (Node *Package Manager*) yang dapat langsung dicocokkan dan diintegrasikan di dalam Express.js [13].

3.6. MySQL

MySQL adalah sistem untuk manajemen basis data relasional secara *open source* yang didukung Oracle (RDBMS) bersumber pada SQL (Structured Query Language). MySQL berjalan di nyaris seluruh *Operation System*, termasuk Mac, Linux, ataupun windows. Walaupun bisa digunakan dalam bermacam aplikasi, MySQL sangat kerap berhubungan dengan aplikasi *website* serta pengembangan server.

MySQL merupakan komponen penting dari *service* LAMP. LAMP merupakan platform pengembangan *website* yang memakai Linux sebagai sistem operasinya, Apache sebagai server *website*, MySQL sebagai sistem *database management* data relasional serta PHP sebagai bahasa scripting berorientasi objek.(Terkadang Perl ataupun Python digunakan sebagai ganti PHP).

MariaDB menggambarkan salah satu basis data yang dapat berjalan pada server yang digunakan untuk menaruh serta memanajemen data. MariaDB tidak jauh berbeda dengan MySQL, sebab MariaDB adalah tipe pengembangan yang sangat *open source* serta *independent* dari MySQL. Semenjak diakuisisinya

MySQL oleh Oracle pada September 2010, Monty Program sebagai pencetus awal kode pada MySQL meningkatkan serta membuat tipe yang lebih independent ialah MariaDB. Sampai saat ini, sudah banyak yang telah melakukan migrasi dari MySQL ke MariaDB, contohnya saja perusahaan raksasa Google dan juga situs besar seperti Wikipedia. Salah satu kelebihan MariaDB adalah karena performannya yang cukup bagus dan tidak berat serta kompatibel dengan MySQL. MariaDB juga kompatibel dengan berbagai macam *platform* seperti LINUX, Windows, MacOS, FreeBSD, Solaris. Dari berbagai aspek yang pada masing-masing menjadi keunggulan kedua DBMS tersebut [14].

3.7. Next.js

Next.js merupakan *framework* atau kerangka kerja dari Reactjs yang digunakan untuk membuat aplikasi *website* pada sisi *client* dan yang saat ini telah diimplementasikan di puluhan ribu *website* di dunia. Next.js saat ini telah digunakan oleh beberapa perusahaan besar seperti Nike, Netflix, dan Playstation. Next.js dibuat karena masalah pre-render secara statis di beberapa halaman yang membuat SEO (*Search Engine Optimization*) dikarenakan file JavaScript harus di load dan menentukan komponen apa yang harus ditampilkan.

Next.js mempunyai beberapa keunggulan seperti *automatic code splitting* yang digunakan sebagai pemecah code agar lebih cepat saat meload halaman, memiliki konsep SSR (*Server Side Rendering*) yang akan membantu untuk memudahkan dalam menggunakan SEO. Next.js juga memiliki direktori pages yang akan sangat memudahkan pada proses *routing* dan dapat dimodifikasi sesuai dengan kebutuhan [15].

3.8. Angular

Angular merupakan *framework* atau kerangka kerja struktural untuk aplikasi web dinamis. Biasanya digunakan sebagai *template* dan memungkinkan penulisan *syntax* di dalam HTML untuk mengekspresikan komponen aplikasi

secara jelas dan ringkas, serta dapat dengan mudah untuk proses *scale-up* atau proses pengembangan dengan pemahaman jika *client-server* atau *front-end* cocok dibangun dengan *declarative programming*. Angular dikembangkan pertama kali oleh Misko Hevery dan Adam Abrons pada tahun 2009. Cohesion pada Angular menghilangkan banyak kode yang harus ditulis. Hal itu terjadi dalam *browser*, menjadikan angular ideal bagi teknologi dan *server* apapun [16].

Angular menggunakan konsep arsitektur yang sudah lama dengan design MVC (*Model-View-Controller*) yang mendapatkan popularitas di industri pengembangan perangkat lunak dan menjadi salah satu legenda desain arsitektur pada kebanyakan perusahaan. Pada dasarnya, model tersebut mewakili pengetahuan bahwa *view* akan bertanggung jawab atas tampilan, sementara *controller* memediasi hubungan antara *model* dan *view*. Namun, konsep-konsep ini sedikit abstrak, dan pola ini mungkin memiliki implementasi yang berbeda tergantung pada bahasa, *platform*, dan tujuan aplikasi. Angular akan mengadopsi MVW (*Model-View-Whatever*). Dengan adanya infrastruktur seperti ini menjelaskan bahwa kerangka kerja memberikan pemisahan yang jelas dari logic dan view serta antara lapisan aplikasi, memberikan modularitas, fleksibilitas, dan *testability* [17].

Angular saat ini dikembangkan oleh Google dan digunakan dalam seluruh *client side* ataupun interaksinya. Angular secara spesifik menawarkan fitur SPA (*Single Page Application*) yang memungkinkan membuat seluruh sisi halaman ke dalam satu halaman saja. Berikut ini beberapa keunggulan dari SPA:

1. Tidak perlunya melakukan *refresh* berkali-kali agar data ke *load*, hal ini dikarenakan realtime data yang terdapat di Angular.
2. Tampilan *user* lebih *friendly*.
3. Dapat bekerja secara *offline*.
4. Lebih stabil dalam mengolah data yang cukup besar.

Angular menerapkan beberapa *design pattern* yang deklaratif serta memiliki beberapa fungsionalitas yang dapat disisipkan secara langsung didalam html secara *directives* [13].

BAB VI. PENUTUP

6.1 Kesimpulan

Berdasarkan hasil dari perancangan, pembahasan, dan pengujian CMS dan *Corona Updates*, maka dapat disimpulkan beberapa hal antara lain:

1. penerapan arsitektur *microservice* pada aplikasi GueSehat dan diimplementasikan pada CMS (*Content Management System*) untuk mengelola layout *service* membuat aplikasi GueSehat lebih dinamis, sehingga tim produk dapat melakukan pengujian melalui A/B Testing untuk membandingkan dua buah tampilan aplikasi mana yang unggul, selain itu dari sisi pengemban tidak perlu mendevlop ulang aplikasi secara terus menerus hanya untuk *update* atau mengubah tampilan/dashboard. Pada fitur *website Corona Updates* memudahkan *user* aplikasi D2D untuk mencari jurnal-jurnal *update* terkait dengan COVID-19.
2. pada pengimplementasian *framework* next js untuk *website* D2D dengan fitur *corona updates* berhasil membantu *user* aplikasi D2D lebih mudah, cepat, dan akurat untuk mencari jurnal-jurnal terkait dengan COVID-19. saat ini aplikasi sudah live pada <https://covid-19.d2d.co.id/>.

6.2 Saran

Berikut adalah saran dari hasil pengembangan dan pengujian CMS dan *website Corona Updates*:

Kebutuhan pada data pada CMS (*Content Management System*) lain yang masih dapat dikembangkan untuk penelitian di masa depan adalah mengatur image yang dapat diubah menjadi *innerHTML* sehingga lebih dinamis dan bisa lebih mudah untuk langsung diterapkan untuk kedepannya.

Kebutuhan pada *website Corona Updates* dapat dikembangkan dengan menambahkan fitur *sorting* sesuai dengan tipe seperti *sort by* tanggal, author, ataupun A-Z.

DAFTAR PUSTAKA

- [1] M. C.R., Iqbal, “Implementasi Klien SIP Berbasis Web,” *J. Tek. Its*, vol. 1, pp. 242–245, 2012.
- [2] M. Z. Siambaton and M. Fakhriza, “Aplikasi *Content Management System* (Cms) Pada Joomla Untuk Membuat *Web Service*,” *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 1, no. 1, pp. 11–13, 2016.
- [3] A. Ulinuha and Y. Irawan, “*Content Management System* (CMS) Untuk Tes Online Mahasiswa Baru Pada Universitas Muria Kudus,” *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 4, no. 1, p. 11, 2014.
- [4] H. Suryotrisongko, “Arsitektur *Microservice* untuk Resiliensi Sistem Informasi,” *Sisfo*, vol. 06, no. 02, pp. 231–246, 2017.
- [5] P. S. Hasugian, “Perancangan *Website* Sebagai Media Promosi Dan Informasi,” *J. Inform. Pelita Nusant.*, vol. 3, no. 1, pp. 82–86, 2018.
- [6] A. R. Sampaio, J. Rubin, I. Beschastnikh, and N. S. Rosa, “Improving *microservice*-based applications with runtime placement adaptation,” *J. Internet Serv. Appl.*, vol. 10, no. 1, 2019.
- [7] “Arsitektur *Microservice*.” [Online]. Available: <https://hackernoon.com/how-microservices-saved-the-internet-30cd4b9c6230>. [Accessed: 03-May-2020].
- [8] G. Munawar, P. N. Bandung, A. Hodijah, and P. N. Bandung, “Analisis Model Arsitektur *Microservice* Pada Sistem Informasi DPLK,” no. October, 2018.
- [9] S. A. Sovia Elinawati, Abulwafa Muhammad, “Perancangan *Content Management System* (Cms) Dengan Studi Kasus E-Bisnis Pada Toko Alya Gorden,” *J. KomTekInfo Fak. Ilmu Komput.*, vol. 2, no. 1, pp. 79–90, 2015.
- [10] I. Yatini B, “Aplikasi pengolahan citra berbasis web menggunakan javascript dan jquery,” *J. Tek.*, vol. 3, no. 3, pp. 1–8, 2014.

- [11] R. FAJRIN, “Pengembangan Sistem Informasi Geografis Berbasis Node.JS untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia,” *J. Inform.*, vol. 11, no. 2, pp. 40–47, 2017.
- [12] H. Shah and T. R. Soomro, “Node. Js Challenges in Implementation,” *Glob. Journals Inc*, vol. 17, no. 2, 2017.
- [13] G. Munawar, “Dalam Pengembangan Web Berbasis Deklaratif,” 2018.
- [14] I. WARMAN and R. RAMDANIANSYAH, “ANALISIS PERBANDINGAN KINERJA QUERY DATABASE MANAGEMENT SYSTEM (DBMS) ANTARA MySQL 5.7.16 DAN MARIADB 10.1,” *J. Teknoif*, vol. 6, no. 1, pp. 32–41, 2018.
- [15] “Learn Next.js.” [Online]. Available: <https://nextjs.org/learn/basics/create-nextjs-app>. [Accessed: 05-May-2020].
- [16] R. Rismanto, P. P. Arhandi, and A. Prasetyo, “Rancang Bangun Aplikasi Ujian Online Real Time Dengan Menggunakan Arsitektur Mean,” *Tekno. Inf.*, vol. 7, no. 2, pp. 149–156, 2016.
- [17] R. Branass, *AngularJS Essentials*. 2014.