

**APLIKASI KKN MAHASISWA UNIVERSITAS LAMPUNG
BERBASIS ANDROID**

(Skripsi)

Oleh

MUHAMMAD APRIANSYAH SETIAWAN



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2018**

ABSTRACT

KKN APPLICATION FOR STUDENT OF LAMPUNG UNIVERSITY ANDROID BASED

BY

MUHAMMAD APRIANSYAH SETIAWAN

KKN in the University of Lampung is a community service activity that must be passed by all students of Lampung University (Unila) who takes bachelor degree program. KKN is a form of student devotion to the people with a cross-scientific and sectorial approaches at a particular time and area. KKN lasts for one to two months and conducted in the village that has been determined by Unila. Students who will participate in must register their self by filling out the form provided by BP-KKN in the KKN website. All information about KKN is on the KKN website. To facilitate students in obtaining all information about KKN and registration, the researcher build a KKN application on android. This application can be used by students to get information about KKN schedule, to register as KKN participant, to get information about location placement and map of village location, to get information about Supervisor in specified area, information regarding the score obtained after complete KKN and can also send and manage student daily activities reports. Through this application, student can be easier to get information about KKN. The result of this research are : (1) functional testing is using Black Box Testing method and Equivalence Partitioning (EP) technique show that application function as expected. Testing and assessment using questionnaires with Likert Scale method, it was obtained an average value of 87.12 % with categorized as "Very Good". The assessment showed that the applications concluded running very well and can be used then by students of the University of Lampung.

Keywords : KKN, *Black Box Testing*, *Equivalence Partitioning*, Likert Scale.

ABSTRAK

APLIKASI KKN MAHASISWA UNIVERSITAS LAMPUNG BERBASIS ANDROID

OLEH

MUHAMMAD APRIANSYAH SETIAWAN

Kegiatan Kuliah Kerja Nyata (KKN) di Universitas Lampung (Unila) merupakan mata kuliah yang wajib diikuti oleh seluruh mahasiswa Universitas Lampung (Unila) yang menempuh program sarjana/strata (S1). KKN adalah bentuk pengabdian mahasiswa kepada masyarakat dengan pendekatan lintas keilmuan dan sektoral pada waktu dan daerah tertentu. Kegiatan KKN biasanya berlangsung antara satu sampai dua bulan dan bertempat di daerah setingkat desa yang telah ditentukan oleh pihak Unila. Bagi mahasiswa yang akan mengikuti kegiatan KKN diwajibkan mendaftarkan diri dengan mengisi form yang telah disediakan oleh pihak BP-KKN yang terdapat di website KKN. Semua info mengenai kegiatan KKN sudah tertera pada website KKN. Akan tetapi untuk memudahkan mahasiswa dalam mendapatkan informasi mengenai KKN dan melakukan pendaftaran, penelitian ingin membangun sebuah aplikasi KKN yang diperuntukkan khusus bagi mahasiswa Universitas Lampung yang berbasis Android. Aplikasi ini dapat digunakan mahasiswa untuk mendapatkan informasi mengenai jadwal KKN, mendaftar sebagai peserta KKN, mendapatkan informasi mengenai lokasi penempatan dan peta lokasi desa, mendapatkan informasi mengenai Dosen Pembimbing Lapangan yang menjadi pembimbing di wilayah yang telah ditentukan, informasi mengenai nilai yang di dapat setelah kegiatan KKN selesai dan dapat juga mengirim dan mengelola laporan kegiatan harian mahasiswa. Dengan ini mahasiswa dapat lebih mudah dalam mendapatkan informasi mengenai KKN. Hasil dari penelitian ini adalah : (1) pengujian fungsional dengan menggunakan metode Black Box Testing dan teknik Equivalence Partitioning (EP) memperoleh hasil sesuai dengan yang diharapkan, (2) pengujian dan penilaian menggunakan kuisisioner dengan perhitungan hasil kuisisioner menggunakan metode Skala Likert yang memperoleh nilai rata-rata sebesar 87.12%, nilai ini termasuk dalam kategori “Sangat Baik”. Dari hasil penilaian yang didapat maka aplikasi ini berjalan sangat baik dan dapat digunakan oleh mahasiswa Universitas Lampung.

Kata Kunci : Kuliah Kerja Nyata (KKN), *Black Box Testing*, *Equivalence Partitioning*, Skala Likert.

**APLIKASI KKN MAHASISWA UNIVERSITAS LAMPUNG
BERBASIS ANDROID**

Oleh

MUHAMMAD APRIANSYAH SETIAWAN

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA KOMPUTER

pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2018**

Judul Skripsi : **APLIKASI KKN MAHASISWA UNIVERSITAS
LAMPUNG BERBASIS ANDROID**

Nama Mahasiswa : **Muhammad Apriansyah Setiawan**

Nomor Pokok Mahasiswa : 1317051037

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

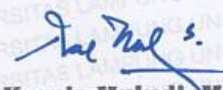
MENYETUJUI

1. Komisi Pembimbing


Dr. Ir. Kurnia Muludi, M.S.Sc.
NIP 19640616 198902 1 001


Anie Rose Irawati, S.T., M.Cs.
NIP 19791031 200604 2 002

2. Ketua Jurusan Ilmu Komputer


Dr. Ir. Kurnia Muludi, M.S.Sc.
NIP 19640616 198902 1 001

MENGESAHKAN

1. Tim Penguji

Ketua

: **Dr. Ir. Kurnia Muludi, M.S.Sc.**



Sekretaris

: **Anie Rose Irawati, S.T., M.Cs.**



Penguji

Bukan Pembimbing : **Ir. Machudor Yusman, M.Kom.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Prof. Warsito, S.Si., D.E.A., Ph.D.

NIP 19710212 199512 1 001

Tanggal Lulus Ujian Skripsi : **01 Maret 2018**

PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul “Aplikasi KKN Mahasiswa Universitas Lampung Berbasis Android” merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang saya terima.

Bandar Lampung, 01 Maret 2018



M. APRIANSYAH SETIAWAN
NPM. 1317015037

RIWAYAT HIDUP



Penulis dilahirkan pada tanggal 01 April 1995 di Bandar Lampung, Lampung sebagai anak keempat dari empat bersaudara dengan Ayah yang bernama Hermansyah B.Sc. dan Ibu bernama Henizam. Penulis menyelesaikan pendidikan dasar di SDN 1 Pelita Bandar Lampung dan selesai pada tahun 2007. Kemudian penulis melanjutkan ke pendidikan menengah pertama di SMPN 25 Bandar Lampung dan diselesaikan penulis pada tahun 2010, kemudian melanjutkan ke pendidikan menengah atas di SMAN 4 Bandar Lampung yang diselesaikan pada tahun 2013.

Pada tahun 2013 penulis terdaftar sebagai mahasiswa di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Selama dalam masa perkuliahan, penulis mengikuti organisasi internal jurusan yaitu Himakom (Himpunan Mahasiswa Jurusan Ilmu Komputer) periode 2013/2014 hingga periode 2015/2016. Selama menjadi mahasiswa beberapa kegiatan yang dilakukan penulis antara lain pada bulan Januari 2016 penulis melaksanakan Kerja Praktik (KP) di Badan Pertanahan Nasional (BPN) Kantah Pesawaran, dan pada bulan Juli 2016 penulis melaksanakan Kuliah Kerja Nyata (KKN) di Desa Nambah Dadi Kecamatan Terbanggi Besar, Lampung Tengah.

PERSEMBAHAN

*Segala puji Syukur atas berkah dan rahmat dari Allah Subhanallah Wata'ala,
Kupersembahkan Skripsi Ini Untuk Orang-Orang Yang Selalu Kuharapkan
Cinta dan Kasih Sayangnya...*

*Teruntuk Papa dan Mama yang tidak pernah memutus doa untuk anak-
anaknya, terimakasih untuk menopang jatuhku dan menuntunku, terimakasih
untuk jadi saksi disetiap langkahku...*

*Teruntuk kakak-kakakku, terima kasih atas teguran, motivasi dan dukungan
dari kalian semua...*

*Teruntuk teman-temanku tercinta, terimakasih untuk selalu ada menunggu dan
menemani...*

*Almamater Tercinta,
UNIVERSITAS LAMPUNG*

MOTTO

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan”
(QS: Al-Insyirah : 5)

“Allah tidak akan membebani seseorang melainkan sesuai dengan kesanggupannya”
(QS: Al-Baqarah : 286)

“A winner is a dreamer, who never gives up”
(Nelson Mandela)

“Do It, Don't Quit”
(Penulis)

SANWACANA

Assallamu'alaikum Warahmatullahi Wabarakatu..

Puji syukur kehadiran Allah Subhanallahu Wata'ala karena atas berkah dan rahmat-Nya penulis dapat menyelesaikan skripsi ini yang berjudul “Aplikasi KKN Mahasiswa Univeristas Lampung Berbasis Android”. Skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

Dalam pelaksanaan dan penyusunan skripsi ini penulis sangat berterima kasih dan memberikan penghargaan yang sedalam-dalamnya kepada seluruh pihak yang membantu penulis menyelesaikan skripsi ini. Penulis ingin mengucapkan terima kasih dengan setulus hati terutama kepada:

1. Saya. Terima kasih kepada saya sendiri yang selalu berusaha dan tidak menyerah, yang tetap berjuang hingga mendapatkan hasil yang memuaskan.
2. Kedua Orangtua tercinta, Mama, Papa, dan kakak-kakakku (Eka Budi Setiawan, S.Farm., Rudi Setiawan, S.Kom., M.Cs, Bambang Setiawan S.Kom.) serta saudara-saudaraku yang selalu ku sayangi dan ku kasihi yang selalu memberikan dukungan, masukan, motivasi dan doanya yang tak terhingga.

3. Bapak Prof. Drs. Warsito, S.Si., D.E.A., Ph.D. selaku Dekan FMIPA Universitas Lampung.
4. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc. selaku ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung, serta pembimbing skripsi atas kesediaannya, kesabaranan dan keikhlasannya untuk memberikan dukungan, bimbingan, nasihat, saran, dan kritik dalam proses penyelesaian skripsi ini.
5. Ibu Anie Rose Irawati, S.T., M.Cs., selaku pembimbing pembantu yang telah ikut memberikan bimbingan, saran dan masukan guna penyempurnaan dalam penulisan skripsi ini.
6. Bapak Ir. Machudor Yusman, M.Kom., selaku dosen pembahas skripsi, yang telah memberikan saran dan masukan guna penyempurnaan dalam penulisan skripsi ini.
7. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu dan pengalaman hidup selama penulis menjadi mahasiswa.
8. Ibu Ade Nora Maela, Mas Irsan, Mas Naufal dan Mas Zai, Mbak Lia yang telah membantu memudahkan segala urusan administrasi penulis di Jurusan Ilmu Komputer.
9. Seluruh keluarga dan saudaraku yang telah memberikan dukungan selama proses perkuliahan yang tidak bisa disebutkan satu persatu.

10. Untuk kekasihku Ananda Putri Kusuma, terimakasih banyak atas semangat dan dukungan nya, terimakasih untuk segala bantuannya, dan terimakasih untuk selalu ada untukku.
11. Sahabat-sahabatku yang sudah seperti keluarga, Dyah Ayu, Alika, Uly, Pina, Ece, Hapis, Dika, Tegar, Adly, Asep, Abdi, Maul, terimakasih banyak untuk selalu ada baik dalam keadaan suka maupun duka.
12. Sahabat-sahabat SMA ku yang masih bertahan hingga sekarang, Ari, Rio, Billy, Ilham, Riska, Liyana. Terima kasih atas support kalian.
13. Rekan-rekan Ilmu Komputer 2013 yang tidak bisa disebutkan satu persatu, terima kasih untuk segala dukungan, bantuan serta kebersamaannya selama ini .
14. Semua pihak yang secara langsung maupun tidak langsung yang telah membantu dalam penyelesaian skripsi ini.

Penulis telah berusaha semaksimal mungkin dalam penulisan skripsi ini untuk mencapai suatu kelengkapan dan kesempurnaan. Penulis juga mengharapkan saran dan kritik yang bersifat membangun dari semua pihak. Akhirnya dengan segala kerendahan hati penulis berharap skripsi ini dapat memberikan manfaat, baik kepada penulis khususnya maupun kepada pembaca pada umumnya.

Bandar Lampung, 01 Maret 2018

Penulis,

Muhammad Apriansyah Setiawan

DAFTAR ISI

	Halaman
DAFTAR ISI	x
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xviii
 I. PENDAHULUAN	 1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
 II. TINJAUAN PUSTAKA	 4
2.1. Kuliah Kerja Nyata.....	4
2.1.1. Bagan Struktur Organisasi Instansi.....	4
2.1.2. <i>Job Description</i>	5
2.2. <i>Extensible Markup Language (XML)</i>	5
2.3. <i>Application Programming Interface (API)</i>	5

2.4. <i>Java Script Object Notation (JSON)</i>	5
2.5. <i>Software Development Kit (SDK)</i>	6
2.6. <i>Metodologi Pengembangan Sistem</i>	6
2.6.1. <i>System Development Life Cycle (SDLC)</i>	6
2.6.2. <i>Unified Modeling Language (UML)</i>	7
2.6.2.1. <i>Usecase Diagram</i>	8
2.6.2.2. <i>Activity Diagram</i>	10
2.6.2.3. <i>Class Diagram</i>	10
2.6.2.4. <i>Sequence Diagram</i>	11
2.6.3. <i>Metode Waterfall</i>	14
2.6.4. <i>Black Box Testing</i>	16
2.6.4.1. <i>Equivalence Partitioning</i>	17
2.6.4.2. <i>Boundary Value Analysis</i>	19
2.6.4.3. <i>Cause-Effect Graphing Techniques</i>	21
2.6.4.4. <i>Comparison Testing</i>	22
2.6.4.5. <i>Automated Testing Tools</i>	25
2.7. <i>Skala Likert</i>	26
2.8. <i>Penelitian Terkait</i>	27
 III. METODOLOGI PENELITIAN	 28
3.1. <i>Waktu dan Tempat Penelitian</i>	28
3.2. <i>Metodologi Penelitian</i>	28
3.2.1. <i>Langkah I</i>	29
3.2.1.1. <i>Identifikasi Masalah</i>	29

3.2.1.2. Studi Literatur.....	29
3.2.2. Langkah II.....	30
3.2.2.1. Analisa Kebutuhan.....	30
3.2.2.2. Desain Sistem.....	31
3.2.2.3. <i>Coding</i>	48
3.2.2.4. <i>Testing</i>	48
IV. HASIL DAN PEMBAHASAN.....	52
4.1. Analisa Kebutuhan.....	52
4.2. Implementasi Sistem.....	52
4.2.1. Halaman Utama.....	53
4.2.2. Halaman Pendaftaran.....	54
4.2.3. Halaman <i>Login</i>	55
4.2.4. Halaman Petunjuk Aplikasi.....	56
4.2.5. Halaman Tentang Aplikasi.....	57
4.2.6. Halaman Menu Mahasiswa/ <i>Home</i>	57
4.2.7. Halaman Info Penempatan.....	58
4.2.8. Halaman Info DPL.....	59
4.2.9. Halaman Lihat Nilai.....	60
4.2.10. Halaman Buat Laporan.....	61
4.2.11. Halaman Laporan Kegiatan.....	62
4.2.12. Halaman Detail Laporan Kegiatan.....	63
4.3. Pengujian Sistem.....	64
4.3.1. Pengujian Internal.....	64

4.3.1.1. Pengujian Fungsional.....	65
4.3.1.1.1. Pengujian Versi Android.....	65
4.3.1.1.2. Pengujian Ukuran Layar.....	66
4.3.1.1.3. Pengujian <i>User Interface</i>	66
4.3.1.1.4. Pengujian Menu Aplikasi.....	67
4.3.2. Pengujian Eksternal.....	69
4.4. Analisa Hasil Kuisisioner.....	71
V. PENUTUP.....	78
5.1. Kesimpulan.....	78
5.2. Saran.....	79
DAFTAR PUSTAKA.....	80
LAMPIRAN.....	82

DAFTAR TABEL

Tabel	Halaman
2.1. Notasi Sequence Diagram.....	12
3.1. Rancangan Daftar Pengujian.....	49
4.1. Pengujian Versi Android.....	65
4.2. Pengujian Ukuran Layar.....	66
4.3. Pengujian User Interface.....	67
4.4. Pengujian Fungsi dari Menu Aplikasi.....	68
4.5. Kriteria Penilaian dan Bobot Nilai.....	69
4.6. Hasil Penilaian Kuisisioner Mahasiswa Unila.....	70
4.7. Kriteria Index Penilaian Hasil Kuisisioner.....	71

DAFTAR GAMBAR

Gambar	Halaman
2.1. Struktur Organisasi Sekretariat KKN Unila.....	4
2.2. Siklus Hidup Pengembangan Sistem.....	7
2.3. Contoh Aktor.....	9
2.4. <i>Use Case</i>	9
2.5. Notasi <i>Activity Diagram</i>	10
2.6. Notasi <i>Class Diagram</i>	11
2.7. Metode <i>Waterfall</i>	14
3.1. Diagram Alir Metodologi Penelitian.....	29
3.2. <i>Use Case Diagram</i>	32
3.3. <i>Activity Diagram</i> Pendaftaran.....	33
3.4. <i>Activity Diagram Login</i>	34
3.5. <i>Activity Diagram</i> Info Penempatan.....	34
3.6. <i>Activity Diagram</i> Informasi DPL dan KDPL.....	35
3.7. <i>Activity Diagram</i> Lihat Nilai.....	35
3.8. <i>Activity Diagram</i> Laporan.....	36
3.9. <i>Activity Diagram</i> Petunjuk.....	37
3.10. <i>Activity Diagram</i> Tentang.....	37

3.11.	<i>Class Diagram</i>	38
3.12.	<i>Sequence Diagram</i> Pendaftaran.....	39
3.13.	<i>Sequence Diagram Login</i>	39
3.14.	<i>Sequence Diagram</i> Info Penempatan.....	40
3.15.	<i>Sequence Diagram</i> Informasi DPL dan KDPL.....	40
3.16.	<i>Sequence Diagram</i> Lihat Nilai.....	41
3.17.	<i>Sequence Diagram</i> Laporan Kegiatan.....	41
3.18.	<i>Sequence Diagram</i> Petunjuk.....	42
3.19.	<i>Sequence Diagram</i> Tentang.....	42
3.20.	Rancangan Halaman Utama.....	43
3.21.	Rancangan Halaman Pendaftaran.....	43
3.22.	Rancangan Halaman <i>Login</i>	44
3.23.	Rancangan Halaman <i>Home</i> Mahasiswa.....	44
3.24.	Rancangan Halaman Info Penempatan.....	45
3.25.	Rancangan Halaman Lihat Nilai.....	45
3.26.	Rancangan Halaman Informasi DPL dan KDPL.....	46
3.27.	Rancangan Halaman Laporan Kegiatan.....	46
3.28.	Rancangan Halaman Buat Laporan.....	47
3.29.	Rancangan Halaman Petunjuk.....	47
3.30.	Rancangan Halaman Tentang.....	48
4.1.	Halaman <i>Home</i>	53
4.2.	Halaman Pendaftaran.....	54
4.3.	Halaman <i>Login</i>	55
4.4.	Halaman Petunjuk Aplikasi.....	56

4.5.	Halaman Tentang Aplikasi.....	57
4.6.	Halaman Menu Mahasiswa.....	58
4.7.	Halaman Info Penempatan.....	59
4.8.	Halaman Info DPL.....	60
4.9.	Halaman Lihat Nilai.....	61
4.10.	Halaman Buat Laporan.....	62
4.11.	Halaman Laporan Kegiatan.....	63
4.12.	Halaman Detail Laporan Kegiatan.....	64
4.13.	Grafik penilaian kuisioner terhadap pernyataan 1.....	72
4.14.	Grafik penilaian kuisioner terhadap pernyataan 2.....	73
4.15.	Grafik penilaian kuisioner terhadap pernyataan 3.....	74
4.16.	Grafik penilaian kuisioner terhadap pernyataan 4.....	75
4.17.	Grafik penilaian kuisioner terhadap pernyataan 5.....	76

DAFTAR LAMPIRAN

Lampiran	Halaman
1. Struktur Organisasi Sekretariat KKN Unila.....	83
2. Kuisioner Pengujian Eksternal.....	87

I. PENDAHULUAN

1.1 Latar Belakang

Universitas Lampung (Unila) merupakan salah satu Perguruan Tinggi Negeri yang terdiri dari 8 fakultas dan 47 jurusan yang terakreditasi. Unila memiliki berbagai macam mata kuliah wajib, salah satunya adalah Kuliah Kerja Nyata (KKN). KKN adalah bentuk kegiatan pengabdian kepada masyarakat oleh mahasiswa dengan pendekatan lintas keilmuan dan sektoral pada waktu dan daerah tertentu. Pelaksanaan kegiatan KKN biasanya berlangsung antara satu sampai dua bulan dan bertempat di daerah setingkat desa yang telah ditentukan oleh pihak Unila. KKN memiliki dampak yang cukup penting bagi mahasiswa, seperti membentuk sikap kepedulian sosial dan tanggung jawab mahasiswa terhadap kemajuan masyarakat, memberikan pengalaman dan keterampilan kepada mahasiswa sebagai kader pembangunan, serta membina mahasiswa agar menjadi seorang inovator dan motivator. Bagi pemerintah daerah dan masyarakat setempat, kegiatan KKN dapat membantu percepatan proses pembangunan serta membentuk kader penerus kegiatan pembangunan.

Mahasiswa yang ingin melakukan kegiatan KKN terlebih dahulu melakukan pendaftaran dengan cara mengisi persyaratan yang telah ditentukan oleh BP-KKN melalui *website* KKN Universitas Lampung. Melalui *website* ini, mahasiswa juga dapat melihat informasi mengenai lokasi penempatan KKN, informasi mengenai

Dosen Pembimbing Lapangan (DPL) dan Ketua Dosen Pembimbing Lapangan, melihat nilai KKN dan mengirim laporan kegiatan KKN.

Sebuah penelitian mengenai pengembangan aplikasi pelaporan kegiatan harian KKN berbasis Android telah dilakukan pada tahun 2016 dalam rangka mengefektifkan proses pengiriman laporan. Namun aplikasi ini hanya dapat digunakan untuk pelaporan kegiatan KKN saja, sedangkan untuk melihat lokasi penempatan, dan info lain, mahasiswa masih harus mengakses melalui *website* (Permana, 2016).

Berdasarkan latar belakang di atas, peneliti ingin mengembangkan aplikasi yang dapat mencakup semua kegiatan yang dilakukan mahasiswa mulai dari kegiatan pendaftaran KKN, melihat informasi lokasi penempatan dengan menampilkan peta lokasi desa tempat mahasiswa tersebut melaksanakan KKN, informasi DPL dan KDPL, melihat nilai hasil KKN serta mengirim laporan kegiatan harian mahasiswa KKN.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini yaitu bagaimana cara mengembangkan sebuah sistem informasi untuk mahasiswa KKN Universitas Lampung berbasis android.

1.3 Batasan Masalah

Batasan masalah penelitian ini adalah sebagai berikut :

1. Aplikasi ini dibangun berbasis android.
2. Aplikasi hanya digunakan oleh Mahasiswa Universitas Lampung.

3. Aplikasi dapat digunakan untuk melakukan pendaftaran KKN.
4. Aplikasi ini dapat mengirim laporan kegiatan harian mahasiswa di lokasi KKN.
5. Aplikasi ini menampilkan informasi lokasi penempatan peserta KKN.
6. Aplikasi ini menampilkan informasi Dosen yang menjadi Pembimbing Lapangan sesuai wilayah.
7. Aplikasi ini menampilkan nilai hasil KKN apabila data nilai sudah diisi oleh pihak BP-KKN.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan mengembangkan sistem informasi yang dapat digunakan oleh mahasiswa KKN Universitas Lampung yang berbasis android.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini adalah sebagai berikut :

1. Mempermudah mahasiswa dalam mendaftarkan diri sebagai peserta KKN.
2. Mempermudah mahasiswa untuk mendapatkan informasi mengenai KKN.

II. TINJAUAN PUSTAKA

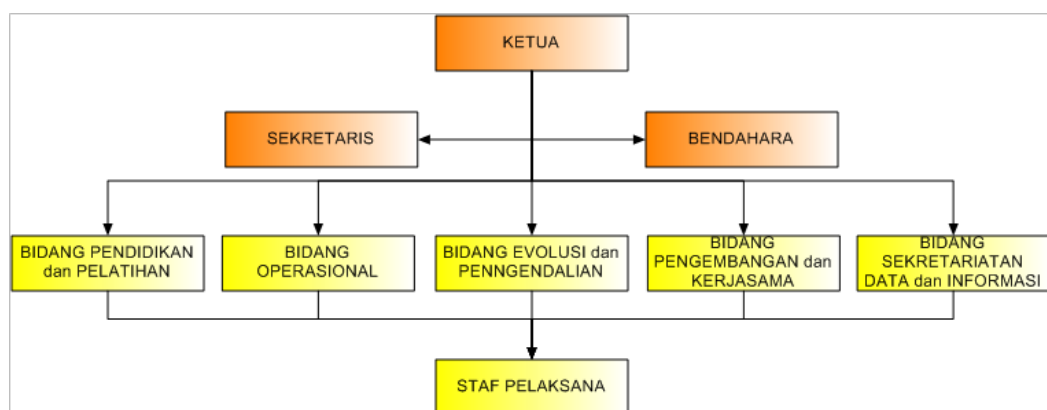
2.1 Kuliah Kerja Nyata

Kuliah Kerja Nyata (KKN) adalah suatu kegiatan intrakulikuler yang memadukan pelaksanaan Tri Dharma Perguruan Tinggi dengan metode pemberian pengalaman belajar dan bekerja kepada mahasiswa dalam kegiatan pemberdayaan masyarakat. Oleh karena itu, KKN diarahkan untuk menjamin keterkaitan antara dunia akademik-teoritik dan dunia empirik-praktis (BP-KKN, 2016).

Paradigma kegiatan KKN harus merespon terhadap tekanan globalisasi saat ini serta peningkatan kualitas hidup masyarakat Indonesia. Oleh sebab itu, program KKN Unila dikembangkan melalui pendekatan pemberdayaan kelompok yang berbasis keluarga, sehingga KKN Unila dinamakan KKN Pemberdayaan Masyarakat dan Pemberdayaan keluarga (BP-KKN, 2016).

1. Bagan Struktur Organisasi Instansi

Struktur organisasi pada Sekretariat KKN Unila dapat dilihat pada Gambar 2.1



Gambar 2.1 Struktur Organisasi Sekretariat KKN Unila (BP-KKN, 2016).

2. Job Description

Job Description Sekretariat KKN Universitas Lampung disajikan pada Lampiran 1.

2.2 Extensible Markup Language (XML)

XML merupakan dasar terbentuknya *web service* yang digunakan untuk mendeskripsikan data. Pada level paling detail *web service* secara keseluruhan dibentuk diatas XML. Fungsi utama dari XML adalah komunikasi antar aplikasi, integrasi data, dan komunikasi aplikasi eksternal dengan *partner* luaran. Dengan standarisasi XML, aplikasi-aplikasi yang berbeda dapat dengan mudah berkomunikasi antar satu dengan yang lain (Deviana, 2011).

2.3 Application Programming Interface (API)

API (*Application Programming Interface*) atau Antarmuka Pemrograman Aplikasi adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu (Ichwan dan Fifin, 2011).

2.4 Java Script Object Notation (JSON)

JSON (*Java Script Object Notation*) adalah format pertukaran data yang bersifat ringan, disusun oleh *Douglas Crockford*. Fokus JSON adalah pada representasi data di *website*. JSON dirancang untuk memudahkan pertukaran data pada situs dan merupakan perluasan dari fungsi-fungsi *javascript* (Chasseur dan Patel, 2013).

2.5 Software Development Kit (SDK)

Android SDK adalah *tools API* yang diperlukan untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman *Java*. Sebagai *platform* aplikasi netral, Android memberi kesempatan bagi semua orang untuk membuat aplikasi yang dibutuhkan, yang bukan merupakan aplikasi bawaan *Handphone/Smartphone* (Safaat, 2011).

2.6 Metodologi Pengembangan Sistem

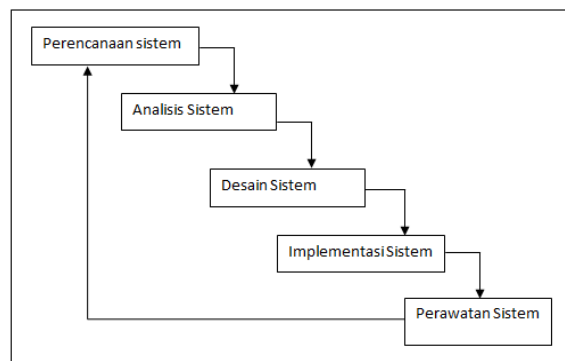
Metodologi pengembangan sistem yang digunakan dalam proses pembuatan Aplikasi KKN Mahasiswa Universitas Lampung, adalah : *System Development Life Cycle* (SDLC) dan desain menggunakan *Unified Modeling Language* (UML).

2.6.1. System Development Life Cycle (SDLC)

Menurut Ladjamudin pada penelitian Rudi, Arief dan Victor (Hermawan, dkk., 2015) bahwa Metode *System Development Life Cycle* atau sering disingkat dengan SDLC merupakan pengembangan yang berfungsi sebagai sebuah mekanisme untuk mengidentifikasi perangkat lunak. Pengembangan sistem informasi yang berbasis komputer dapat merupakan tugas kompleks yang membutuhkan banyak sumber daya dan dapat memakan waktu untuk menyelesaikannya. Proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan, dan dipelihara. Daur atau siklus hidup dari pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-

langkah di dalam tahapan tersebut dalam proses pengembangannya (Hermawan, dkk., 2015).

Tahapan dalam pengembangan sistem dinamakan *System Development Life Cycle* (SDLC) karena pada setiap tahapan sistem dikerjakan secara berurut menurun dari perencanaan, analisis, desain, implementasi dan perawatan. Disajikan dalam Gambar 2.2 (Hermawan, dkk., 2015).



Gambar 2.2 Siklus Hidup Pengembangan Sistem (Hermawan, dkk., 2015).

2.6.2. *Unified Modeling Language (UML)*

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual seperti kata Braun pada Haviluddin (2011). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Haviluddin, 2011).

Sejarah UML sendiri terbagi dalam dua fase; sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain

berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi (Haviluddin, 2011).

Fase kedua; dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh *Object Management Group* (OMG) maka pengembangan UML dimulai pada akhir tahun 1994 ketika *Grady Booch* dengan metode OOD (*Object-Oriented Design*), *Jim Rumbaugh* dengan metode OMT (*Object Modelling Technique*) mereka ini bekerja pada *Rasional Software Corporation* dan *Ivar Jacobson* dengan metode OOSE (*Object-Oriented Software Engineering*) yang bekerja pada perusahaan *Objectory Rasional* (Haviluddin, 2011).

UML dideskripsikan oleh beberapa diagram, yaitu sebagai berikut (Fowler, 2004).

2.6.2.1. Use Case Diagram

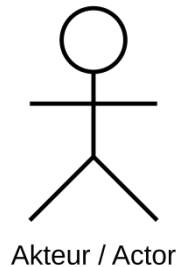
Use case Diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case* diagram lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case* diagram merepresentasikan sebuah interaksi antara aktor dengan sistem (Fowler, 2004).

Komponen-komponen dalam *use case* diagram (Fowler, 2004) :

1. Aktor

Pada dasarnya aktor bukan bagian dari *use case* diagram, namun untuk dapat terciptanya suatu *use case* diagram diperlukan aktor, dimana aktor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem yang dibuat. Sebuah aktor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari

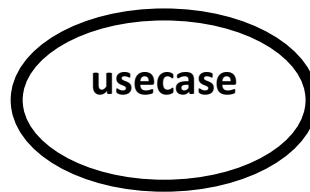
sistem atau keduanya menerima dan memberi informasi pada sistem. Aktor hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Aktor digambarkan dengan *stick pan* seperti yang terdapat pada Gambar 2.3.



Gambar 2.3 Contoh Aktor (Fowler, 2004)

2. *Use Case*

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti kegunaan sistem yang dibangun. Bentuk *use case* dapat terlihat pada Gambar 2.4.



Gambar 2.4 *Use Case* (Fowler, 2004)

Terdapat 4 (empat) relasi yang ada pada *use case* diagram (Fowler, 2004) :

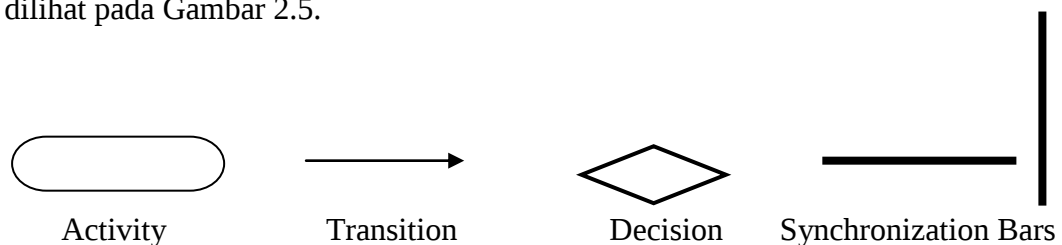
1. *Association*, menghubungkan *link* antar *element*.
2. *Generalization*, disebut juga pewarisan (*inheritance*), sebuah elemen dapat merupakan spesialisasi dari *element* lainnya.
3. *Dependency*, sebuah *element* bergantung dalam beberapa cara ke *element* lainnya.
4. *Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

Tipe relasi yang mungkin terjadi pada *use case* diagram (Fowler, 2004) :

1. <<*include*>>, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
2. <<*extends*>>, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan peringatan.
3. <<*communicates*>>, merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara aktor dan *use case*.

2.6.2.2. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya (Fowler, 2004). Notasi Activity Diagram dapat dilihat pada Gambar 2.5.



Gambar 2.5 Notasi Activity Diagram (Haviluddin, 2011)

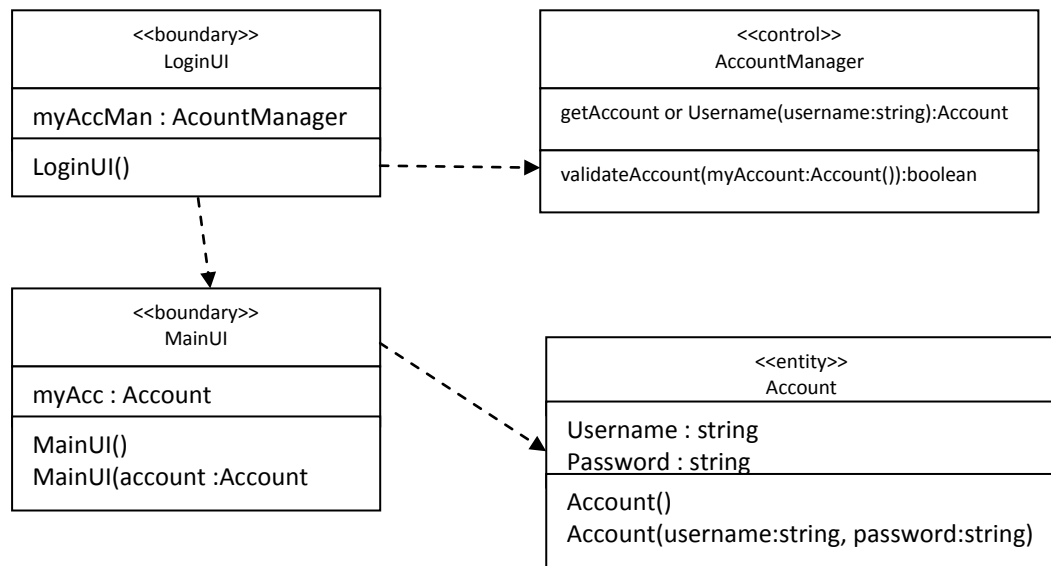
2.6.2.3. Class Diagram

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan

merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat (Haviluddin, 2011).

Class memiliki tiga area pokok (Haviluddin, 2011) :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

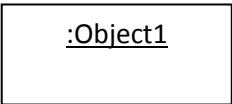
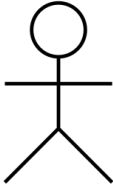


Gambar 2.6 Notasi *Class Diagram*

2.6.2.4. *Sequence Diagram*

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. (Fowler, 2014). Notasi *Sequence Diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Notasi *Sequence Diagram* (Meildy, 2014)

Simbol	Nama	Keterangan
	<i>Object</i>	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma
	<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom. Simbol <i>Actor</i> sama dengan simbol pada <i>Actor Use Case Diagram</i> .
	<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah obyek

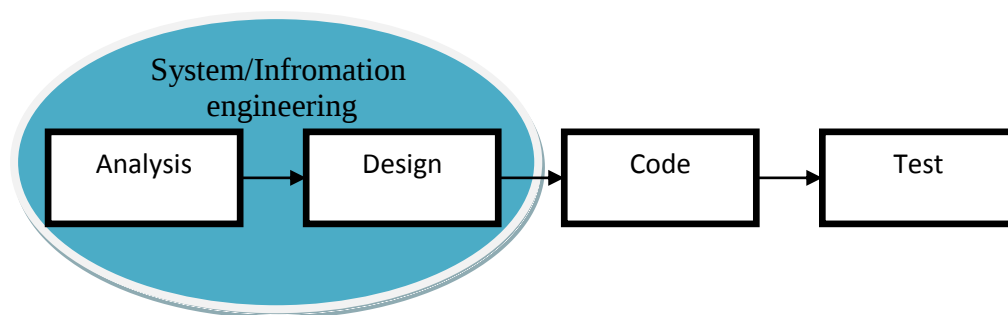
Tabel 2.1 Notasi *Sequence* Diagram (lanjutan)

	<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.
Message 	<i>Message</i>	<i>Message</i> , digambarkan dengan anak panah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara objek-objek.

2.6.3. Metode *Waterfall*

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, proses yang berjalan terus mengalir ke bawah fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian (Pressman, 2001).

Tahapan yang dilakukan dalam pengembangan aplikasi ini sesuai dengan Metode *Waterfall* dapat dilihat pada Gambar 2.5.



Gambar 2.7 Metode *Waterfall* (Pressman, 2001).

Tahapan pengembangan sistem dengan Metode *Waterfall* dijelaskan sebagai berikut (Pressman, 2001) :

1.6 Analisis Kebutuhan Aplikasi (*Anaylisis*)

Analisa kebutuhan sistem atau aplikasi merupakan tahapan pertama yang menjadi dasar proses pembuatan sistem. Pada tahap ini dilakukan pencarian dan pengumpulan semua kebutuhan yang diperlukan untuk menunjang kelengkapan sistem atau aplikasi, kemudian mendefinisikan semua kebutuhan yang dipenuhi dalam perangkat lunak atau aplikasi yang dibuat.

1.7 Desain Aplikasi (*Design*)

Desain aplikasi merupakan tahap perancangan sistem atau aplikasi yang meliputi penyusunan proses, data, aliran proses, dan pemenuhan kebutuhan

sesuai dengan hasil analisa kebutuhan. Dokumentasi desain aplikasi yang dihasilkan dari tahapan ini adalah *Use Case Diagram* dan *Activity Diagram*.

1.8 Penerapan Desain dan Penulisan Kode Program (*Code*)

Penulisan kode program merupakan tahap penerjemahan desain sistem yang telah dibuat ke dalam bentuk perintah-perintah yang dimengerti komputer dengan menggunakan bahasa pemrograman. Penelitian ini menggunakan bahasa pemrograman *Java* dan *Android Studio* sebagai *software* pengembangan aplikasinya. Pada tahap ini, penulis menerjemahkan design kedalam bahasa pemrograman sehingga didapatkan suatu aplikasi yang diinginkan sesuai yang sudah dirancang, sehingga didapatkan suatu *file installer* dengan ekstensi *apk*.

1.9 Pengujian Aplikasi (*f*)

Pengujian aplikasi dilakukan untuk memastikan bahwa sistem yang dibuat telah sesuai dengan desain dan semua fungsi dapat dipergunakan dengan baik tanpa ada kesalahan sesuai dengan kebutuhan pengguna. Pengujian aplikasi ini menggunakan metode *Blackbox Testing*. Pengujian dilakukan secara menyeluruh tanpa melihat struktur internal aplikasi atau komponen yang diuji. *Blackbox Testing* berfokus pada kebutuhan fungsional aplikasi yang berdasarkan pada spesifikasi kebutuhan aplikasi tersebut.

1.10 Penerapan Aplikasi dan Perawatan

Pada tahapan ini, aplikasi sudah siap untuk diterapkan pada perangkat *mobile* dan siap digunakan sesuai dengan tujuan dibuatnya aplikasi ini. Perawatan, perbaikan dan pengembangan aplikasi dilakukan untuk menjaga kualitas dan kestabilan aplikasi.

2.6.4. *Black box Testing*

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program (Mustaqbal dkk., 2015).

Black Box Testing bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing* (Mustaqbal dkk., 2015).

Black Box Testing cenderung untuk menemukan hal-hal berikut (Mustaqbal dkk., 2015) :

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan - pertanyaan berikut (Mustaqbal dkk., 2015) :

1. Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?
2. Input seperti apa yang dapat menjadi bahan kasus uji yang baik?
3. Apakah sistem sensitif pada input-input tertentu?
4. Bagaimana sekumpulan data dapat diisolasi?

2.6.4.1. *Equivalence Partitioning*

Equivalence partitioning adalah metode *black box testing* yang membagi domain masukan dari suatu program ke dalam kelas-kelas data, dimana *test cases* dapat diturunkan (BCS, 2001).

Equivalence partitioning berdasarkan pada premis masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas, menurut spesifikasi dari komponen tersebut, yang diperlakukan sama (ekuivalen) oleh komponen tersebut. Dapat juga diasumsikan bahwa masukan yang sama menghasilkan respon yang sama pula (BCS, 2001).

Nilai tunggal pada suatu partisi ekuivalensi diasumsikan sebagai representasi dari semua nilai dalam partisi. Hal ini digunakan untuk mengurangi masalah yang tidak mungkin untuk testing terhadap tiap nilai masukan (lihat prinsip *testing: testing* yang komplit tidak mungkin). Petunjuk pelaksanaan dalam melakukan *equivalence partitioning*, adalah sebagai berikut (BCS, 2001) :

1. Jika masukan mempunyai jenjang waktu tertentu, maka defisiniskan kategori valid dan tak valid terhadap jenjang masukan tersebut.

2. Jika masukan membutuhkan nilai tertentu, definisikan kategori valid dan tak valid.
3. Jika masukan membutuhkan himpunan masukan tertentu, definisikan kategori valid dan tak valid.
4. Jika masukan adalah *boolean*, definisikan kategori valid dan tak valid.

Sedangkan beberapa kombinasi yang mungkin dalam partisi ekuivalensi, adalah (BCS, 2001) :

1. Nilai masukan yang valid dan tak valid.
2. Nilai *numeric* yang negatif, positif atau nol.
3. *String* yang kosong atau tidak kosong.
4. Daftar (*list*) yang kosong atau tidak kosong.
5. *File* data yang ada dan tidak, yang dapat dibaca / ditulis atau tidak.

Misalkan, terdapat data terpelihara untuk sebuah aplikasi perbankan otomatis. *User* dapat mengaksesnya dari komputer pribadinya dengan menyediakan *password* 6 digit, dan mengikuti serangkaian perintah *keyword* yang mengakses berbagai fungsi perbankan. *Software* yang digunakan untuk aplikasi perbankan menerima data dalam bentuk (Ayuliana, 2009) :

1. *Area code* – *blank* atau 3 digit nomor
2. *Prefix* – 3 digit nomor yang tidak diawali oleh 0 atau 1
3. *Suffix* – 4 digit nomor
4. *Password* – 6 digit alphanumerik
5. *Commands* – "*check*", "*deposit*", "*bill pay*", dsb

Kondisi input yang dihubungkan dengan setiap elemen data untuk aplikasi perbankan dapat dispesifikasikan sebagai (Ayuliana, 2009) :

1. *Area code* : kondisi input, *Boolean* – *area code* boleh ada maupun tidak
2. Kondisi input, *Range* – nilai didefinisikan antara 200 dan 999, dengan beberapa pengecualian khusus (misal : tidak ada nilai > 905) dan syarat (misal : seluruh *area code* memiliki angka 0 atau 1 pada posisi digit ke-2)
3. *Prefix* : kondisi input, *Range* – nilai yang dispesifikasikan > 200
4. *Suffix* : kondisi input, *Value* – sepanjang 4 digit
5. *Password* : kondisi input, *Boolean* – *Password* boleh ada maupun tidak
kondisi input, *Value* – 6 string karakter
6. *Command* : kondisi input, *Set* – mengandung perintah-perintah yang ada diatas

Aplikasikan panduan untuk derivasi dari *class-class* yang ekuivalen, kasus uji untuk setiap domain input data item dapat dibentuk dan dieksekusi. Kasus uji dipilih sehingga sejumlah atribut dari *equivalence class* dieksekusi sekali saja (Ayuliana, 2009).

2.6.4.2. Boundary Value Analysis

Boundary value analysis adalah salah satu teknik *black box testing* yang melakukan pengujian pada batas atas dan batas bawah nilai yang diisikan pada aplikasi. Beberapa prinsip yang mendasari pada *boundary value analysis* (BVA) yaitu (Mustaqbal dkk., 2015) :

1. Banyak kesalahan terjadi pada kesalahan masukan.
2. BVA mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input.

3. BVA merupakan komplemen dari *equivalence partitioning*. Lebih pada memilih elemen-elemen di dalam kelas ekivalen pada bagian sisi batas dari kelas.
4. Contoh :
 - a. Untuk rentang yang dibatasi a dan b maka uji (a-1), a, (a+1), (b-1), b, (b+1).
 - b. Jika kondisi input mensyaratkan sejumlah n nilai maka uji dengan sejumlah (n-1), n dan (n+1) nilai.
 - c. Aplikasikan dua aturan sebelumnya pada kondisi output (buat *table* pengujian hasil outputnya untuk nilai maksimal dan minimal). d.
 - d. Jika struktur data internal dari program memiliki cakupan (misal: ukuran *buffer*, batas *array*) gunakan data input yang menguji batas cakupan.

Secara umum, aplikasi BVA dapat dikerjakan secara *generic*. Bentuk dasar implementasi BVA adalah untuk menjaga agar satu *variable* berada pada nilai nominal (normal atau rata-rata) dan mengijinkan *variable* lain diisikan dengan nilai ekstrimnya. Nilai yang digunakan untuk menguji keekstriman data adalah (Mustaqbal dkk., 2015) :

Min ----- minimal

Min+ ----- di atas

Nom ----- rata-rata

Max- ----- tepat di bawah maksimum

Max ----- Maksimum

Sebagai contoh, misalnya dientrikan data tanggal. Data tanggal memiliki tiga *variable* yaitu tanggal, bulan dan tahun. Maka untuk ketiga *variable* tersebut, dapat diambil kondisi berikut (Mustaqbal dkk., 2015) :

$$1 \leq \text{tanggal} \leq 31$$

$$1 \leq \text{bulan} \leq 12$$

$$1812 \leq \text{tahun} \leq 2022$$

Maka untuk setiap entri data di luar angka di atas menampilkan pesan “Tanggal yang anda isikan salah” (Mustaqbal dkk., 2015).

2.6.4.3. Cause-Effect Graphing Techniques

Cause Effect Graphing merupakan teknik pengujian yang diciptakan oleh *Bill Elmendorf* dari IBM pada tahun 1973. Teknik ini membuat kasus uji dengan menggunakan grafik *Boolean Cause-Effect* dalam menguji fungsionalitas suatu perangkat lunak. *Cause Effect Graph* membentuk hubungan antara *output* dan *input* pada sebuah grafik *Boolean* dimana *input* sebagai *cause* dan *output* sebagai *effect* (Karuniawati, 2015).

Tahapan yang dilakukan dalam membuat kasus uji dengan metode *Cause Effect Graphing*, diantaranya (Karuniawati, 2015) :

1. Mengidentifikasi *Cause* dan *Effect* dari spesifikasi yang ada. *Cause* merupakan kondisi *input* dan *Effect* merupakan kondisi *output* atau transformasi sebuah sistem
2. Mengambarkan hubungan antar *cause* dan *effect* menggunakan grafik *cause* dan *effect*

3. Mengidentifikasi *constraint* untuk menjelaskan kombinasi antar *cause* dan antar *effect* yang tidak mungkin karena adanya batasan lingkungan.
4. Mengubah grafik *cause & effect* menjadi *Decision Table*

2.6.4.4. Comparison Testing

Dalam beberapa situasi (seperti : *aircraft avionic, nuclear power plant control*) dimana keandalan suatu *software* amat kritis, beberapa aplikasi sering menggunakan *software* dan *hardware* ganda (*redundant*). Ketika *software redundant* dibuat, tim pengembangan *software* lainnya membangun versi independen dari aplikasi dengan menggunakan spesifikasi yang sama. Setiap versi dapat diuji dengan data uji yang sama untuk memastikan seluruhnya menyediakan *output* yang sama. Kemudian seluruh versi dieksekusi secara parallel dengan perbandingan hasil *real-time* untuk memastikan konsistensi (Ayuliana, 2009).

Dianjurkan bahwa versi independen suatu *software* untuk aplikasi yang amat kritis harus dibuat, walaupun nantinya hanya satu versi saja yang digunakan dalam sistem. Versi independen ini merupakan basis dari teknik *black box testing* yang disebut *comparison testing* atau *back-to-back testing* (Ayuliana, 2009).

Ketika *multiple* implementasi dari spesifikasi yang sama telah diproduksi, kasus uji didesain dengan menggunakan teknik *black box* yang lain (misalkan *equivalence partitioning*) disediakan sebagai *input* untuk setiap versi dari *software*. Jika setiap *output*-nya sama, diasumsikan implementasinya benar, jika tidak, setiap versi di periksa untuk menentukan jika kerusakan terdapat pada satu atau lebih versi yang bertanggung jawab atas perbedaan tersebut (Ayuliana, 2009).

Jika setiap spesifikasi dari seluruh versi telah dibuat dalam kesalahan, maka seluruh versi merefleksikan kesalahan. Sebagai tambahan, jika setiap versi independent memberikan hasil identik, tetapi salah, ujicoba hasil dan kondisi gagal untuk mendeteksi kesalahan (Ayuliana, 2009).

Ujicoba Untuk Sistem *Real Time*

Karakteristik khusus untuk sistem *realtime* memberikan tantangan tersendiri ketika ujicoba dilaksanakan. Ketergantungannya dengan waktu, sifat alami dari beberapa aplikasi yang tidak sinkron, menambah kesulitan baru dan potensial sebagai elemen untuk ujicoba dengan waktu beragam. Tidak hanya ujicoba *whitebox* maupun *blackbox*, tetapi juga ketepatan waktu pengiriman data dan pemrosesan paralel. Contohnya *software realtime* yang mengontrol mesin *fotocopy*, yang dapat menerima interupsi dari operator (berupa penekanan tombol ‘*RESET*’ atau ‘*DARKEN*’) dengan tanpa kesalahan ketika mesin sedang berjalan (‘*COPYING*’ state). Operator yang sama menginterupsi, ketika mesin berada dalam posisi ‘*jamned*’ (Ayuliana, 2009).

Sebagai tambahan, keterkaitan antara *software real time* dengan perangkat keras pendukungnya juga dapat menyebabkan masalah dalam ujicoba. Ujicoba *software* harus mempertimbangkan kesalahan perangkat keras yang disebabkan karena pemrosesan *software*. Belum ada uji kasus yang komprehensif yang dikembangkan untuk sistem *realtime*, tetapi 4 langkah strategi berikut dapat dilaksanakan (Ayuliana, 2009) :

1. *Task Testing*, yaitu dengan mengujicobakan setiap *task* secara independen.

Dalam hal ini metode *whitebox* dan *blackbox testing* dapat digunakan untuk menemukan kesalahan logika dan kesalahan fungsional, tetapi untuk

kesalahan ketepatan waktu dan perilaku *software* (*timing or behavioral errors*), tidak dapat terdeteksi.

2. *Behavioral Testing*, dengan menggunakan model sistem dengan *CASE tool*, memungkinkan untuk mensimulasikan perilaku sistem *realtime* dan menentukannya sebagai konsekuensi dari peristiwa eksternal. Aktivitas analisis ini dapat dilaksanakan sebagai dasar untuk desain kasus uji yang diadakan ketika sebuah sistem *realtime* berhasil dibuat. Dengan menggunakan teknik yang sesuai (seperti *equivalence partitioning*), *event* dikategorikan (misalnya : *interrupts, control signals, data*), misalkan *event* pada sebuah mesin fotokopi dapat berupa interupsi dari user ('*reset counter*'), interupsi mekanikal ('*paper jammed*'), interupsi sistem ('*toner low*'), dan kesalahan bentuk ('*overheated*'). Setiap kesalahan yang terjadi diuji secara individual, dan perilaku *executable* sistem diperiksa. Perilaku *software* diperiksa untuk menentukan apakah terdeteksi kesalahan perilaku sistem.
3. *Intertask Testing*, ketika sebuah kesalahan dari *individual task* berhasil diisolasi, ujicoba berlanjut kepada kesalahan pada waktu yang terkait. *Task* yang tidak sinkron yang berkomunikasi dengan *task* lainnya diuji dengan beberapa data dan pemrosesan untuk menentukan apakah kesalahan antar *task* terjadi. Sebagai tambahan *task* yang berkomunikasi via antrian pesan atau penyimpanan data, diujikan untuk menemukan kesalahan ukuran penyimpanan.

4. *System Testing*, *software* dan *hardware* telah disatukan dan diujikan dalam uji sistem sebagai satu kesatuan. Uji ini dilakukan untuk menemukan kesalahan pada *software/hardware interface*.

2.6.4.5. *Automated Testing Tools*

Dikarenakan ujicoba *software* menghabiskan sekitar 40% dari total usaha yang diperlukan untuk sebuah proyek pengembangan *software*, maka *tools* yang dapat mengurangi waktu uji sangat dibutuhkan. Dengan melihat manfaat potensialnya, maka dibentuklah generasi pertama dari *automated test tools*. Miller mendeskripsikannya menjadi beberapa kategori, diantaranya (Ayuliana, 2009) :

1. *Static Analyzer*, program sistem analisis ini mendukung untuk pembuktian dari pernyataan tanpa bukti statis, perintah-perintah yang lemah dalam struktur program dan format.
2. *Code Auditors*, sebuah filter dengan kegunaan khusus yang digunakan untuk memeriksa kualitas *software* untuk memastikan bahwa *software* tersebut telah memenuhi standar pengkodean minimum
3. *Assertion Processors*, sistem preprocessor/postprocessor ini digunakan untuk memberitahukan programmer dengan menyediakan klaim, yang disebut *assertion*, yaitu mengenai suatu perilaku program yang benar-benar ditemukan saat pelaksanaan program riil
4. *Test File Generators*, merupakan pembangun *processor*, dan dipenuhi dengan definisi awal nilai, *file* masukan yang serupa untuk program yang sedang diujikan.

5. *Test Data Generators*. merupakan sistem analisis otomatis yang membantu pengguna aplikasi dalam memilih data uji yang menyebabkan program berperilaku khusus
6. *Test Verifiers, tool* ini mengukur cakupan uji internal, terkadang berhubungan dengan uji struktur control dari objek uji, dan melaporkan cakupan nilai untuk para ahli jaminan kualitas
7. *Test Harnesses. Tools* ini mendukung pemrosesan uji coba dengan (1) menginstal program kandidat dalam lingkungan uji, (2) berikan input data, dan (3) simulasikan dengan menggunakan *stubs* perilaku dari modul-modul subordinat
8. *Output Comparators, tool* ini membuatnya sangat memungkinkan untuk membandingkan sekumpulan output dari suatu program dengan sekumpulan output dari proses sebelumnya(yang telah diarsipkan) untuk menentukan perbedaan diantara keduanya

2.6.5. Skala Likert

Metode ini merupakan metode penskalaan pernyataan sikap yang menggunakan distribusi *respons* sebagai dasar penentuan nilai skalanya. Nilai skala setiap pernyataan tidak ditentukan oleh derajat *favourable* masing-masing, tetapi ditentukan oleh distribusi *respons* setuju dan tidak setuju dari sekelompok responden yang bertindak sebagai kelompok uji coba (Azwar, 2011).

Skala Likert, yaitu skala yang berisi lima tingkat preferensi jawaban dengan pilihan sebagai berikut : 1= tidak baik; 2 = kurang baik; 3 = cukup baik atau

netral; 4 = baik; 5 = sangat baik. Persentase penilaian berdasarkan kriteria *skala likert* diperoleh dengan rumus aritmatika *mean*, yaitu (Djarwanto, 1996) :

$$P = \frac{X_i}{n \times N} \times 100 \% \quad \dots\dots\dots(1)$$

Keterangan :

P = Persentase pernyataan

X_i = Nilai kunatitatif total

n = Jumlah responden

N = Nilai kategori pernyataan terbaik

Dan penentuan interval tiap kategori menggunakan rumus sebagai berikut :

$$I = \frac{100\%}{K} \quad \dots\dots\dots(2)$$

Keterangan :

I = Interval

K = Kategori Interval

2.6.6. Penelitian Terkait

Hasil penelitian yang relevan dengan penelitian ini adalah :

1. Pengembangan Sistem Pelaporan Kegiatan KKN Berbasis Android (Pratama, 2016). Aplikasi ini digunakan oleh mahasiswa dan hanya untuk pengumpulan laporan kegiatan mahasiswa KKN Universitas Lampung. Untuk aplikasi ini ada baiknya bila ditambahkan fitur untuk melakukan pendaftaran agar lebih efisien.
2. Analisis Pengelompokan Mahasiswa KKN Berdasarkan Kriteria Jenis Kelamin, Fakultas dan Sekolah (Muwisnawangsa, 2016). Penelitian ini mengenai pengoptimalan sistem informasi KKN dan menganalisis pengelompokkan mahasiswa KKN berdasarkan *criteria* jenis kelamin, fakultas dan sekolah. Pada sistem ini masih berbasis *web*, ada baiknya jika sistem ini dikembangkan berbasis Android.

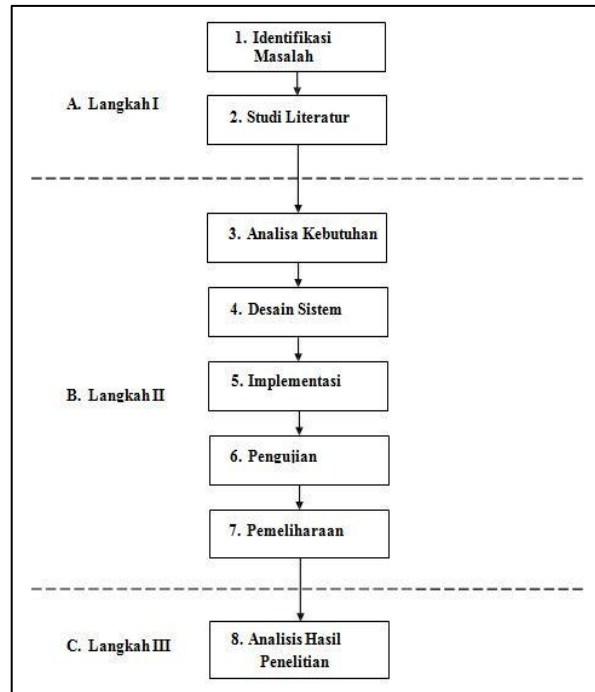
III. METODOLOGI PENELITIAN

3.1. Waktu dan Tempat Penelitian

Penelitian ini dilakukan di Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang berada di Jalan Soemantri Brojonegoro No. 1 Gedung Meneng, Bandar Lampung. Penelitian ini dilaksanakan mulai bulan Februari 2017 sampai Mei 2017.

3.2. Metodologi Penelitian

Metodologi penelitian adalah suatu proses yang digunakan untuk memecahkan suatu masalah secara logis, dimana memerlukan data-data untuk mendukung terlaksananya suatu penelitian. Metode penelitian yang dilakukan berupa studi pustaka, studi literatur, pengembangan sistem, dan analisis hasil penelitian. Penelitian dilakukan berdasarkan diagram alir metodologi penelitian yang terdapat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

Penjelasan dari diagram alir metodologi penelitian pada Gambar 3.1 adalah sebagai berikut.

3.2.1. Langkah I

3.2.1.1. Tahap pertama yang dilakukan yaitu identifikasi masalah yang merupakan tahapan dasar dimana pada tahapan ini dilakukan pengidentifikasian dan penganalisaan terhadap permasalahan yang ada. Tahapan ini menghasilkan perumusan masalah, tujuan penelitian, manfaat penelitian, dan juga batasan-batasan permasalahan yang telah dijelaskan pada Bab I.

3.2.1.2. Tahap kedua yang dilakukan adalah studi literatur. Studi literatur merupakan kegiatan yang meliputi mencari secara literatur,

melokalisasi, dan menganalisis dokumen yang berhubungan dengan masalah yang diteliti. Dokumen bisa berupa teori-teori dan bisa pula hasil penelitian yang telah dilakukan mengenai permasalahan yang diteliti. Sebagaimana yang terdapat pada Bab II.

3.2.2. Langkah II (Metode Pengembangan Sistem)

Memasuki langkah kedua pada metode penelitian ini adalah tahap pengembangan sistem. Metode pengembangan sistem yang dipilih dalam penelitian ini adalah model pengembangan sistem *Waterfall*. Model *waterfall* atau yang sering disebut model *system development life cycle* menunjukkan pengembangan perangkat lunak secara berurutan dan sistematis dimulai dari tahap analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Tahapan dari model *Waterfall* dapat dilihat pada Gambar 2.7 sebelumnya.

Tahap - tahap yang dilakukan dalam pengembangan sistem ini adalah :

3.2.2.1. Analisa Kebutuhan

Tahap pertama yang dilakukan pada langkah kedua yaitu menganalisa kebutuhan dasar sistem baik kebutuhan fungsional maupun nonfungsional.

Kebutuhan fungsional yaitu sebagai berikut :

1. Data persyaratan dalam pendaftaran menjadi peserta KKN
2. Data mahasiswa yang telah melakukan KKN Periode I 2017
3. Data letak geografis desa yang menjadi tempat KKN

Kebutuhan nonfungsional yaitu sebagai berikut :

1. Perangkat Keras (*Hardware*)

- *Processor : Intel ® Core™ i3-3217U CPU @ 1.80GHz (4CPU)*
- *RAM : 8 GB*
- *Harddisk : 500 GB*

2. Perangkat Lunak (*Software*)

- *Operating System : Windows 7 Home Premium 64-bit*
- *Android Studio*
- *XAMPP*
- *Web Browser Google Chrome dan Mozilla Firefox*
- *StarUML*
- *Adobe Photoshop CS6*

3.2.2.2. Desain Sistem

Pada tahap ini dilakukan perancangan desain sistem dan perancangan *interface*.

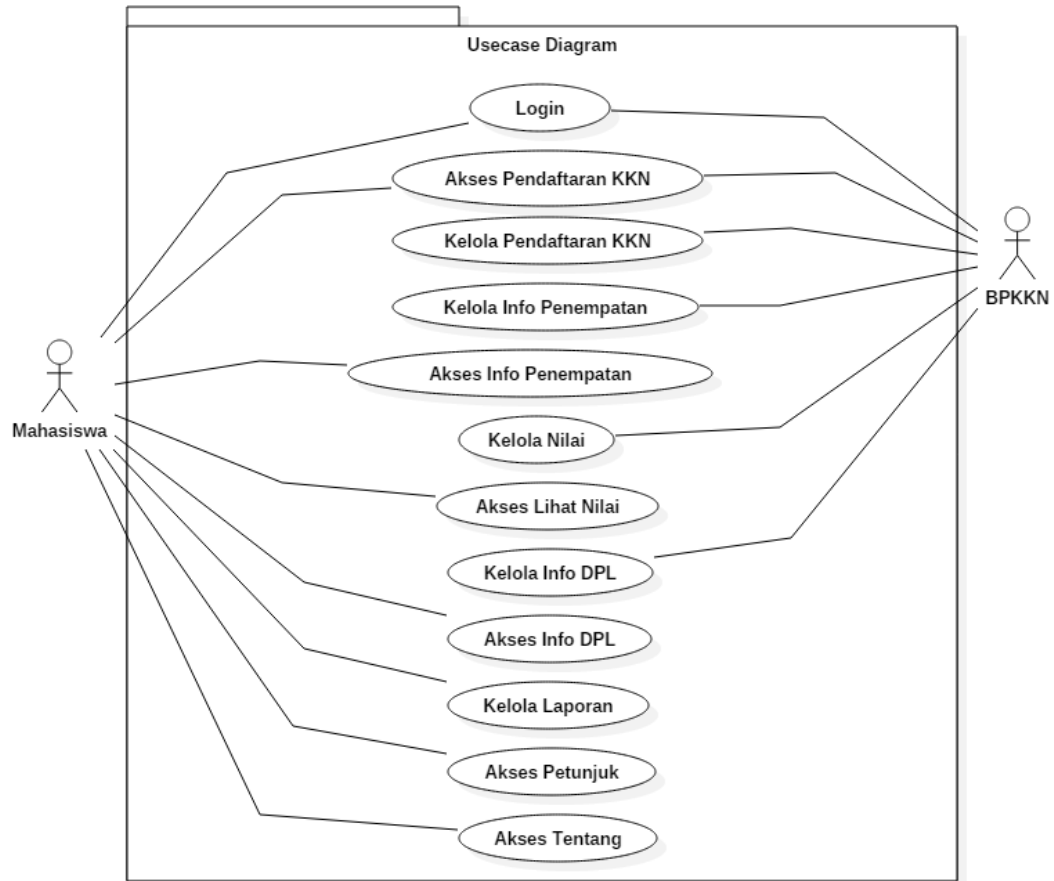
Perancangan desain sistem menggunakan *Unified Modelling Language* (UML).

Diagram-diagram UML yang dibuat dalam penelitian ini adalah sebagai berikut.

1. *Use Case Diagram*

Use Case diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna, sehingga pembuatan *use case* ini dititik beratkan pada fungsionalitas

yang ada pada sistem. Pada sistem ini pengguna dapat melakukan 7 interaksi. *Use case diagram* ini dapat dilihat pada Gambar 3.2



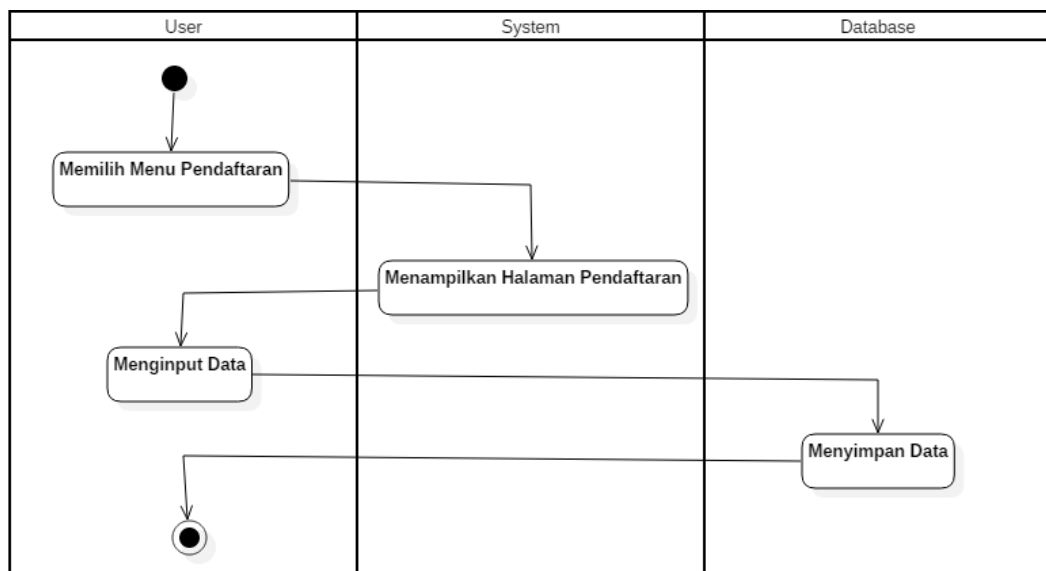
Gambar 3.2 *Use Case Diagram*

2. *Activity Diagram*

Activity diagram digunakan untuk menggambarkan serangkaian aliran dari aktivitas, berguna juga untuk mendeskripsikan aktivitas yang dibentuk dalam satu operasi sehingga dapat juga untuk aktivitas lainnya. Pada sistem ini terdapat 8 *activity diagram* yaitu sebagai berikut.

a. *Activity Diagram Pendaftaran*

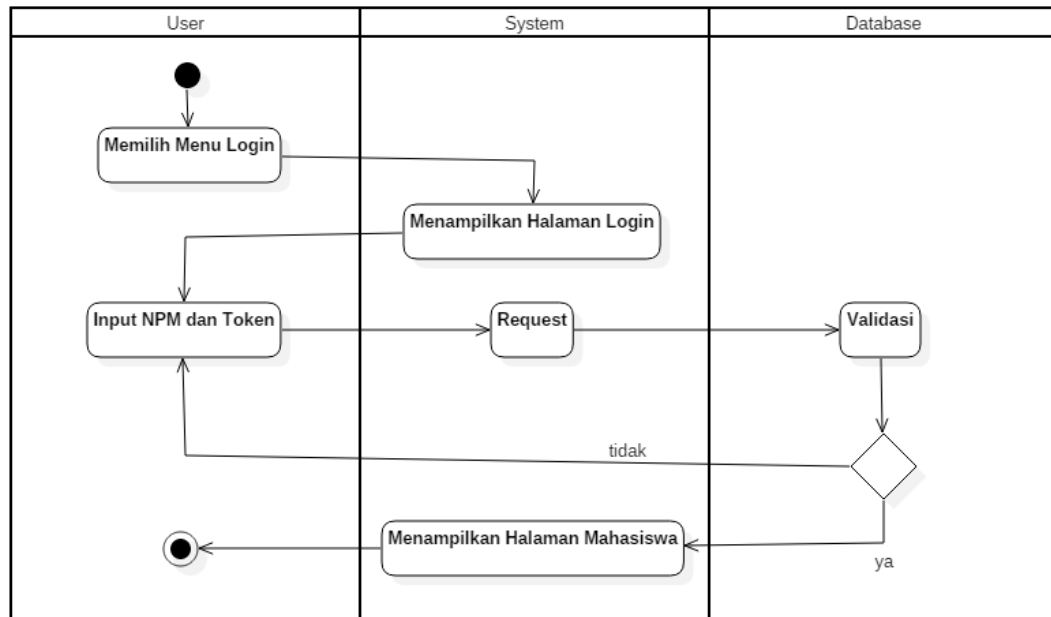
Pada *Activity diagram* pendaftaran berawal dari mahasiswa melakukan pendaftaran sebagai peserta KKN ke sistem dengan memasukkan data pribadi, data orang tua dan data orang yang dapat dihubungi lalu menekan tombol daftar. *Activity diagram* pendaftaran dapat dilihat pada Gambar 3.3.



Gambar 3.3 *Activity Diagram Pendaftaran*

b. *Activity Diagram Login*

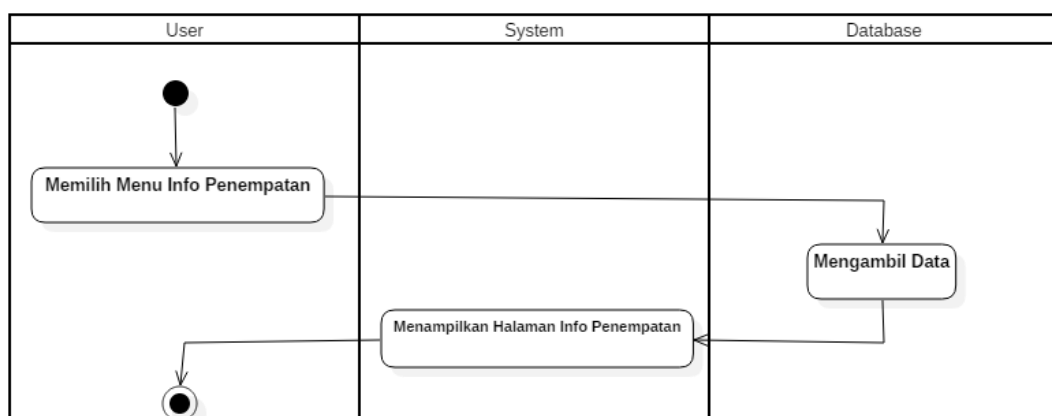
Pada *activity diagram login* mahasiswa memasukkan NPM sebagai *username* dan token yang didapat sebagai *password*. *Activity diagram login* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Activity Diagram Login

c. Activity Diagram Info Penempatan

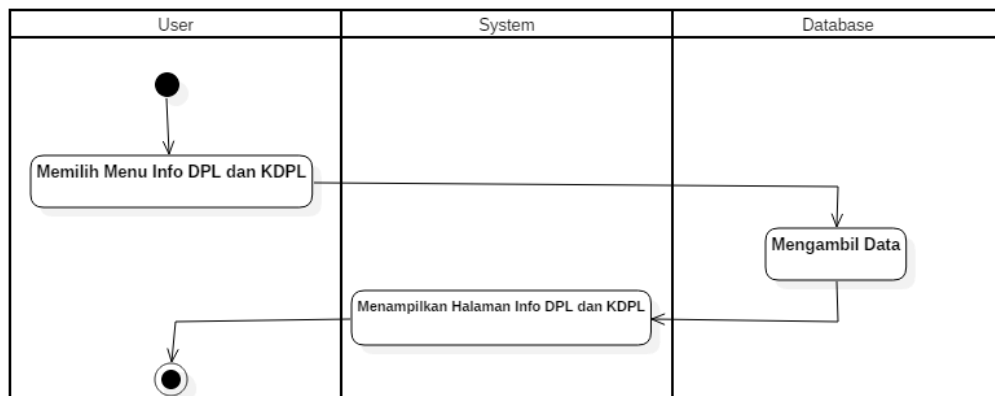
Pada *activity* diagram info penempatan, mahasiswa dapat melihat info mengenai lokasi penempatan, *Activity* diagram info penempatan dapat dilihat pada Gambar 3.5.



Gambar 3.5 Activity Diagram Info Penempatan

d. *Activity Diagram Informasi DPL dan KDPL*

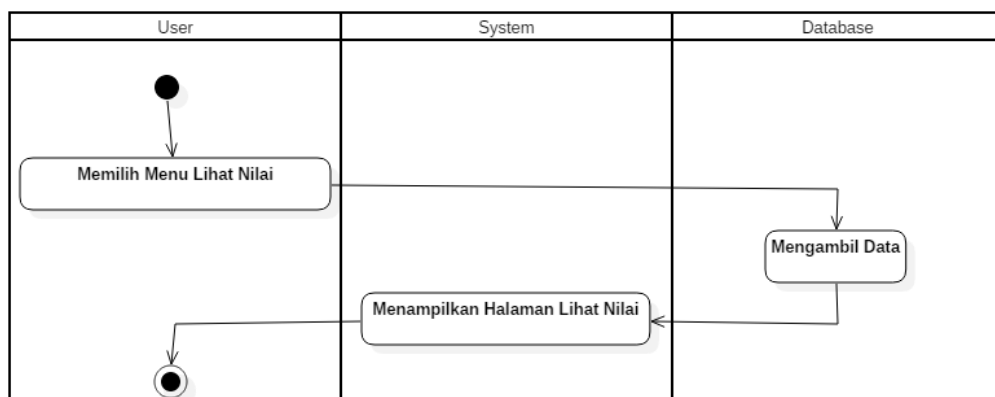
Pada *activity* informasi DPL dan KDPL, mahasiswa dapat melihat informasi mengenai dosen yang menjadi pembimbing lapangan sesuai dengan lokasi KKN mahasiswa. *Activity* diagram Informasi DPL dan KDPL dapat dilihat pada Gambar 3.6



Gambar 3.6 *Activity Diagram Informasi DPL dan KDPL*

e. *Activity Diagram Lihat Nilai*

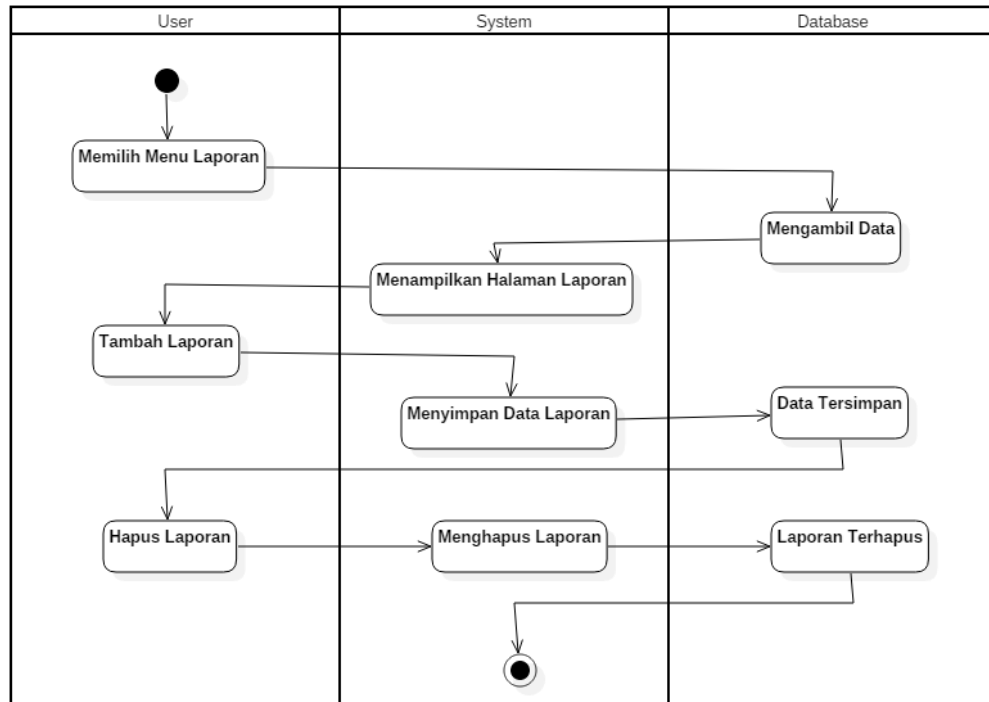
Pada *activity* lihat nilai, mahasiswa dapat melihat nilai yang didapat nilai - nilai sudah diisi. *Activity* diagram lihat nilai dapat dilihat pada Gambar 3.7.



Gambar 3.7 *Activity Diagram Lihat Nilai*

f. *Activity Diagram Laporan*

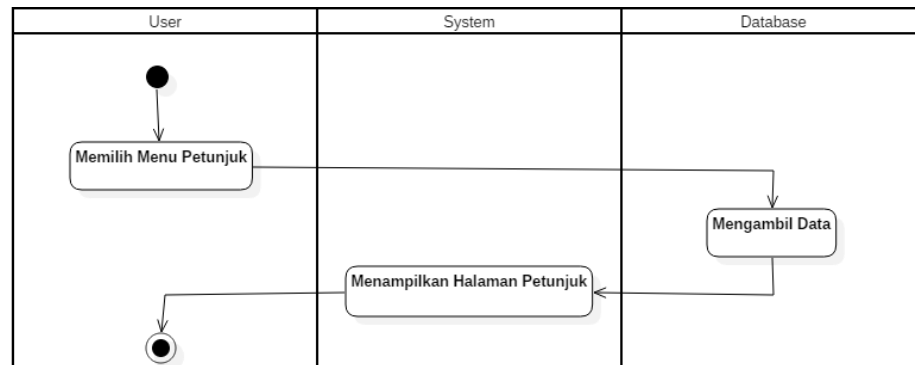
Pada *activity diagram* laporan, mahasiswa dapat melihat laporan yang sudah pernah dikirim. *Activity diagram* laporan dapat dilihat pada Gambar 3.8.



Gambar 3.8 *Activity Diagram Laporan*

g. *Activity Diagram Petunjuk*

Pada *activity diagram* petunjuk, mahasiswa dapat melihat petunjuk penggunaan aplikasi. *Activity diagram* petunjuk dapat dilihat pada Gambar 3.9.

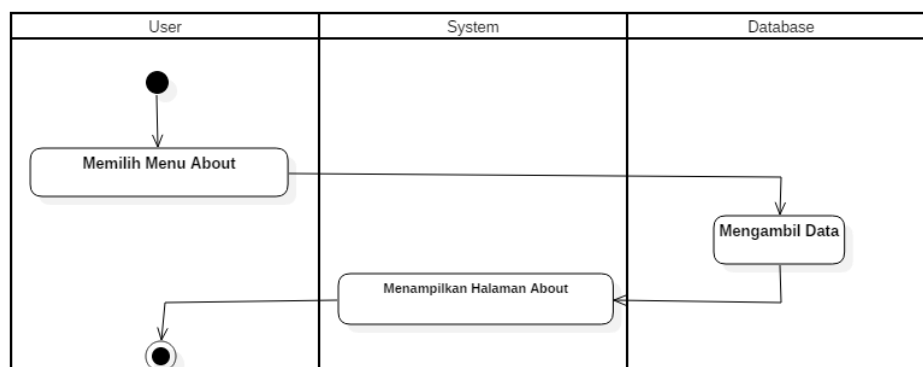


Gambar 3.9 Activity Diagram Petunjuk

h. Activity Diagram Tentang

Pada *activity* diagram tentang, mahasiswa dapat melihat tentang aplikasi.

Activity diagram tentang dapat dilihat pada Gambar 3.10.



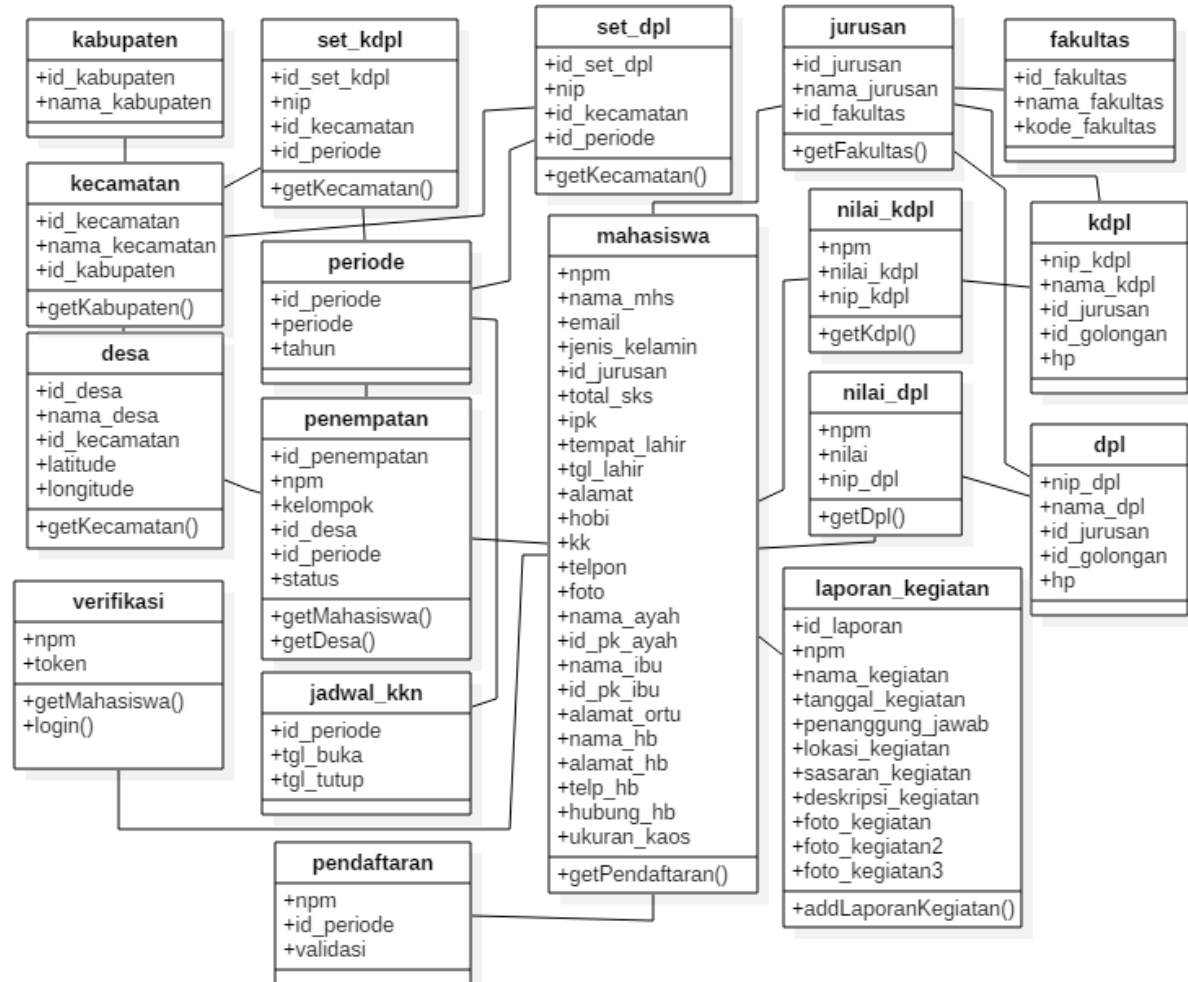
Gambar 3.10 Activity Diagram Tentang

3. Class Diagram

Class Diagram menggambarkan *boundary*, *controller*, dan *entity*. *Boundary* mewakili unsur-unsur perangkat lunak seperti layar, halaman HTML, atau antarmuka sistem yang berinteraksi dengan *user*. *Controller* berfungsi sebagai

perekat antara *boundary* elemen dan *entity* elemen, sebagai elemen proses atau hanya sebagai pengendali. *Entity* adalah elemen yang bertanggung jawab menyimpan data atau informasi. *Class Diagram* ditunjukkan pada Gambar

3.11.



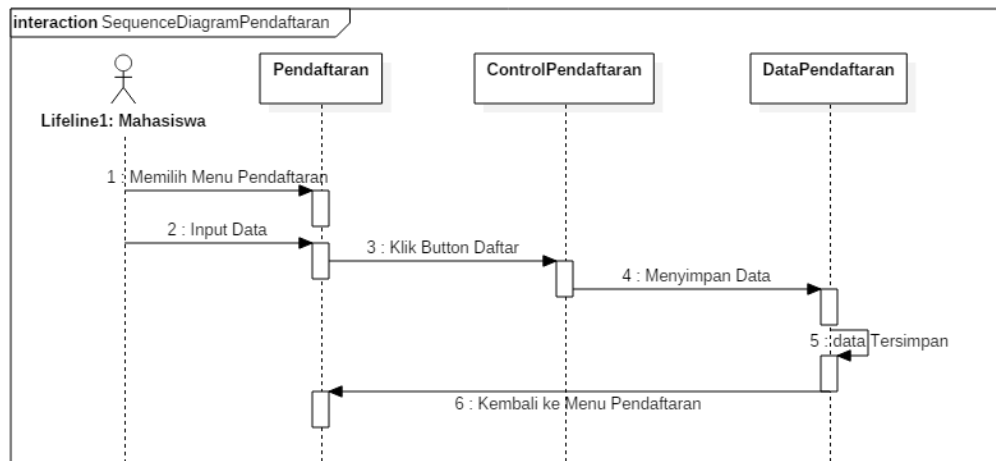
Gambar 3.11 Class Diagram

4. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan arus pekerjaan aplikasi, pesan yang disampaikan dan bagaimana elemen-elemen di dalamnya bekerja sama dari waktu ke waktu untuk mencapai suatu hasil. Pada aplikasi terdapat 8 desain *Sequence Diagram*.

a. *Sequence Diagram Pendaftaran*

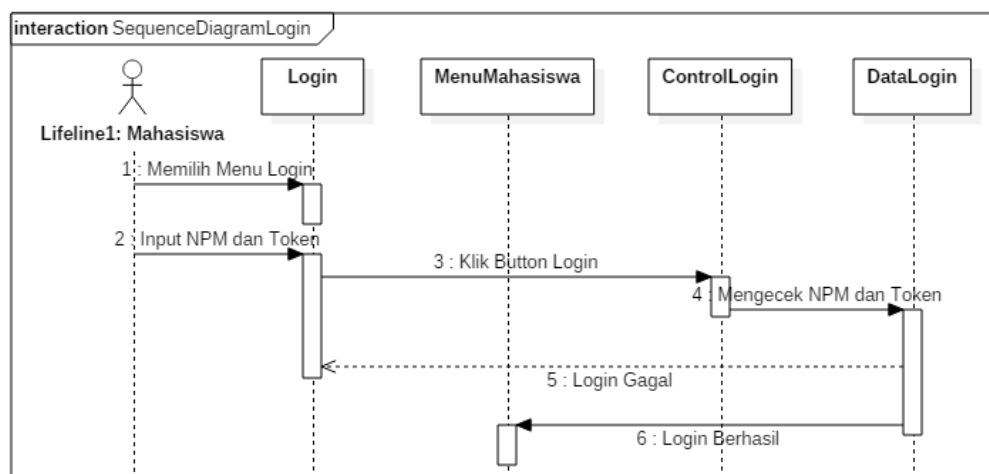
Pada *sequence diagram pendaftaran*, menggambarkan arus pekerjaan pada menu pendaftaran. *Sequence diagram pendaftaran* dapat dilihat pada Gambar 3.12.



Gambar 3.12 *Sequence Diagram Pendaftaran*

b. *Sequence Diagram Login*

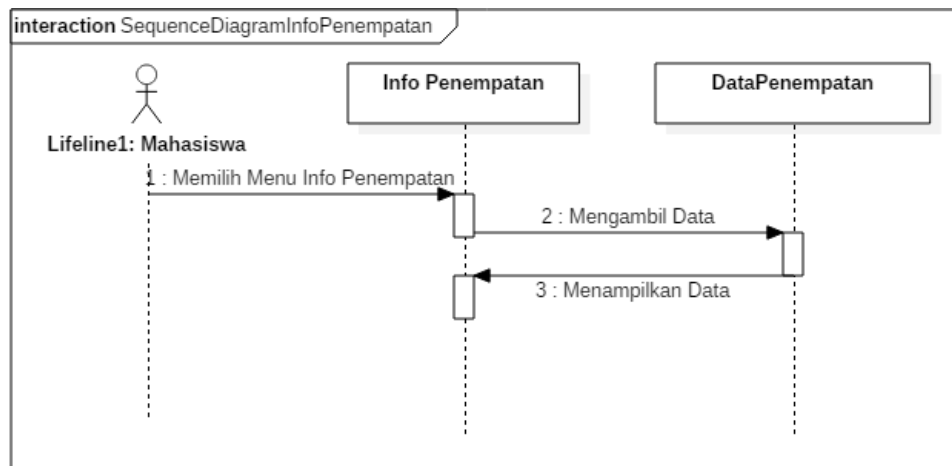
Pada *sequence diagram login*, menggambarkan arus pekerjaan pada menu login. *Sequence diagram login* dapat dilihat pada Gambar 3.13.



Gambar 3.13 *Sequence Diagram Login*

c. *Sequence Diagram Info Penempatan*

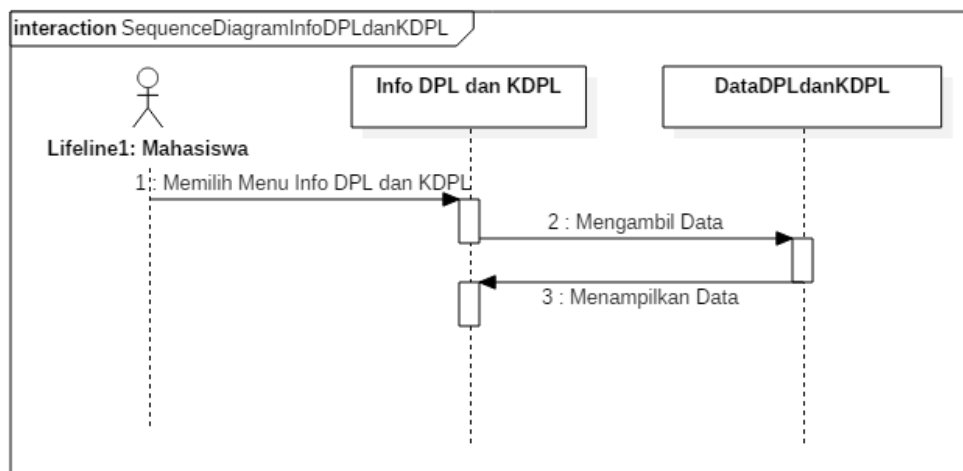
Pada *sequence diagram* info penempatan, menggambarkan arus pekerjaan pada menu info penempatan. *Sequence diagram* info penempatan dapat dilihat pada Gambar 3.14.



Gambar 3.14 *Sequence Diagram Info Penempatan*

d. *Sequence Diagram Informasi DPL dan KDPL*

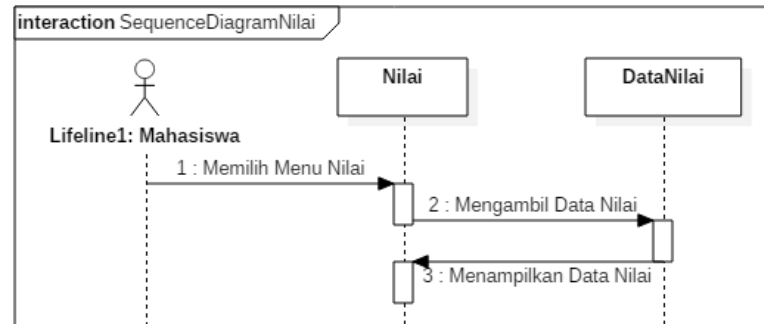
Pada *sequence diagram* informasi DPL dan KDPL, menggambarkan arus pekerjaan pada menu informasi DPL dan KDPL. *Sequence diagram* informasi DPL dan KDPL dapat dilihat pada Gambar 3.15.



Gambar 3.15 *Sequence Diagram Pengumuman DPL dan KDPL*

e. *Sequence Diagram Lihat Nilai*

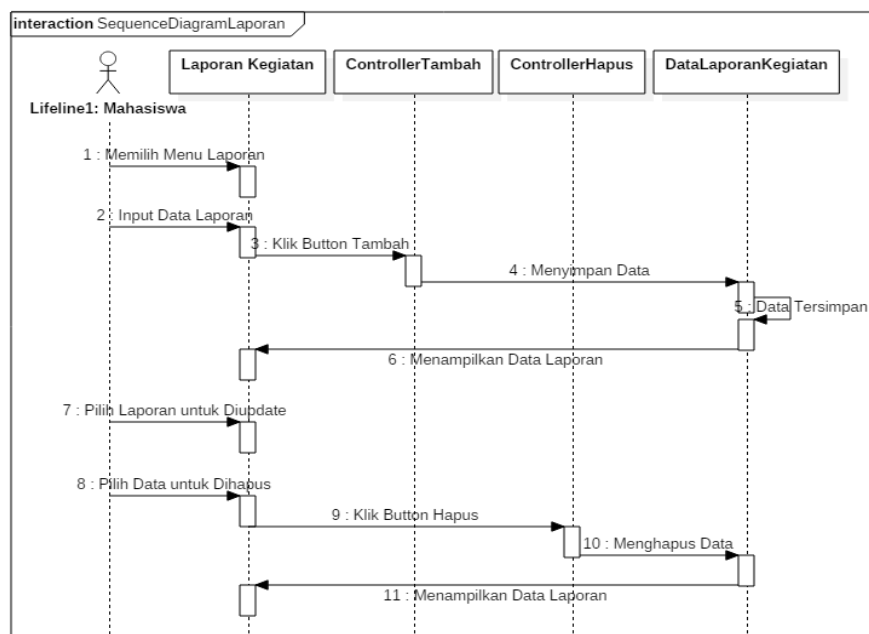
Pada *sequence diagram* lihat nilai, menggambarkan arus pekerjaan pada menu lihat nilai. *Sequence diagram* lihat nilai dapat dilihat pada Gambar 3.16.



Gambar 3.16 *Sequence Diagram Lihat Nilai*

f. *Sequence Diagram Laporan Kegiatan*

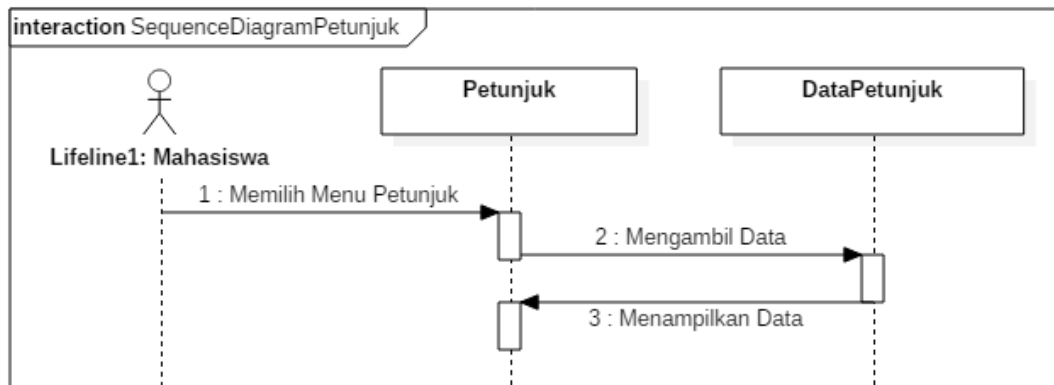
Pada *sequence diagram* laporan kegiatan, menggambarkan arus pekerjaan pada menu laporan kegiatan. *Sequence diagram* laporan kegiatan dapat dilihat pada Gambar 3.17.



Gambar 3.17 *Sequence Diagram Laporan Kegiatan*

g. *Sequence Diagram Petunjuk*

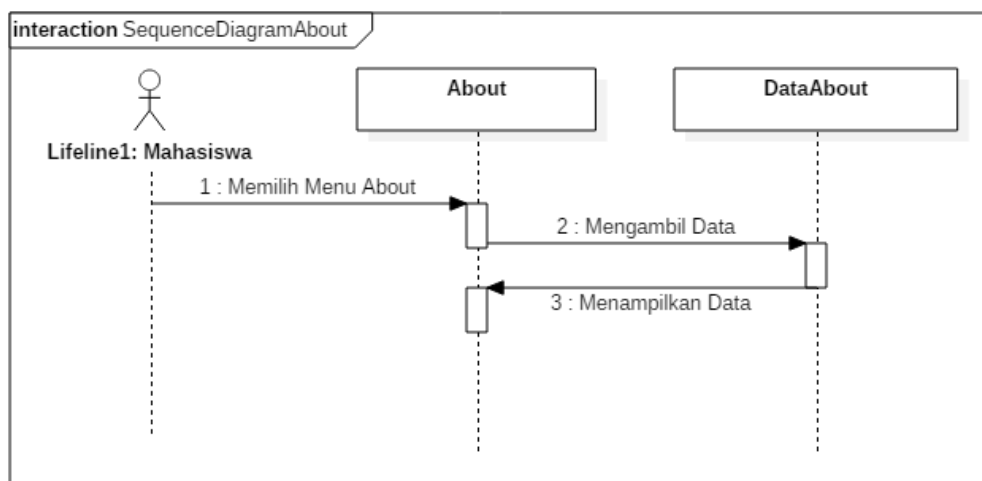
Pada *sequence diagram petunjuk*, menggambarkan arus pekerjaan pada menu petunjuk. *Sequence diagram petunjuk* dapat dilihat pada Gambar 3.18.



Gambar 3.18 *Sequence Diagram Petunjuk*

h. *Sequence Diagram About*

Pada *sequence diagram about*, menggambarkan arus pekerjaan pada menu tentang. *Sequence diagram about* dapat dilihat pada Gambar 3.19.



Gambar 3.19 *Sequence Diagram Tentang*

5. Perancangan Desain *User Interface*

Perancangan desain *user interface* pada Sistem Informasi Mahasiswa KKN Universitas Lampung adalah sebagai berikut.

1. Halaman Utama

Halaman Utama dapat dilihat pada Gambar 3.20.



Gambar 3.20 Rancangan Halaman Utama

2. Halaman Pendaftaran

Halaman Pendaftaran dapat dilihat pada Gambar 3.21



Gambar 3.21 Rancangan Halaman Pendaftaran

3. Halaman *Login*

Halaman *Login* dapat dilihat pada Gambar 3.22.



Gambar 3.22 Rancangan Halaman *Login*

4. Halaman Menu Mahasiswa

Halaman Menu Mahasiswa dapat dilihat pada Gambar 3.23.



Gambar 3.23 Rancangan Halaman *Home* Mahasiswa

5. Halaman Info Penempatan

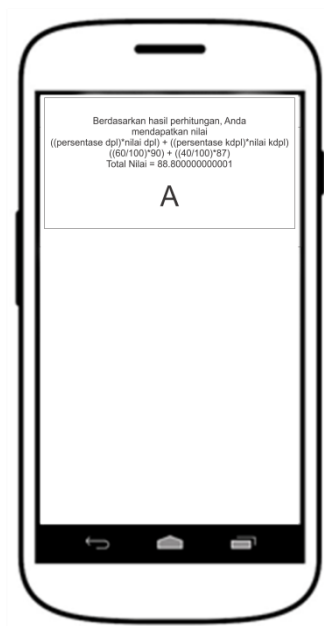
Halaman Info Penempatan dapat dilihat pada Gambar 3.24.



Gambar 3.24 Rancangan Halaman Info Penempatan

6. Halaman Lihat Nilai

Halaman Lihat Nilai dapat dilihat pada Gambar 3.25.



Gambar 3.25 Rancangan Halaman Lihat Nilai

7. Halaman Informasi DPL dan KDPL

Halaman Informasi DPL dan KDPL dapat dilihat pada Gambar 3.26.



Gambar 3.26 Rancangan Halaman Informasi DPL

8. Halaman Laporan Kegiatan

Halaman Laporan Kegiatan dapat dilihat pada Gambar 3.27.



Gambar 3.27 Rancangan Halaman Laporan Kegiatan

9. Halaman Buat Laporan

Halaman Buat Laporan dapat dilihat pada Gambar 3.28.

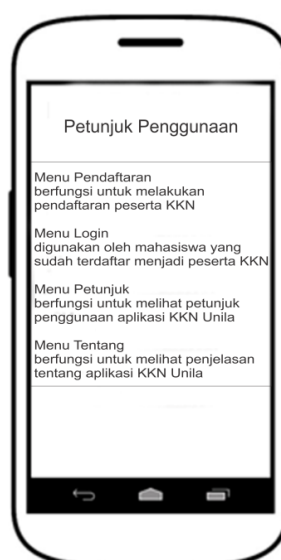


The image shows a mobile application screen for creating a report. It features several input fields for data entry: 'Nama Kegiatan', 'Waktu Kegiatan' (with a date format 'dd-MM-yyyy' and a calendar icon), 'Penanggung Jawab Kegiatan', 'Lokasi Kegiatan', 'Sasaran Kegiatan', and 'Deskripsi Kegiatan'. Below these fields is a large area with a plus sign and an upload icon, likely for attaching a file or image. The screen is framed by a black border representing the phone's bezel, and the bottom shows standard Android navigation icons.

Gambar 3.28 Rancangan Halaman Buat Laporan

10. Halaman Petunjuk

Halaman Petunjuk dapat dilihat pada Gambar 3.29.

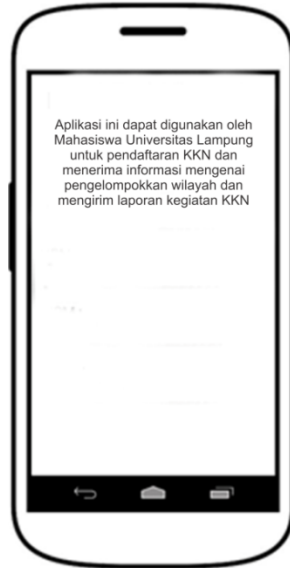


The image shows a mobile application screen titled 'Petunjuk Penggunaan'. It contains a list of instructions for using the application: 'Menu Pendaftaran berfungsi untuk melakukan pendaftaran peserta KKN', 'Menu Login digunakan oleh mahasiswa yang sudah terdaftar menjadi peserta KKN', 'Menu Petunjuk berfungsi untuk melihat petunjuk penggunaan aplikasi KKN Unila', and 'Menu Tentang berfungsi untuk melihat penjelasan tentang aplikasi KKN Unila'. The screen is framed by a black border representing the phone's bezel, and the bottom shows standard Android navigation icons.

Gambar 3.29 Rancangan Halaman Petunjuk

11. Halaman Tentang

Halaman Tentang dapat dilihat pada Gambar 3.30.



Gambar 3.30 Rancangan Halaman Tentang

3.2.2.3. Coding

Tahap *coding* ini merupakan tahapan dalam pengimplementasian desain yang sudah dirancang ke dalam bahasa yang dapat dikenali oleh komputer. Dilakukan oleh *programmer* yang menerjemahkan transaksi yang diminta oleh *user*. Tahapan ini merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer dimaksimalkan dalam tahapan ini.

3.2.2.4. Testing

Pengujian yang dilakukan pada pengembangan aplikasi ini yaitu pengujian *black box* dengan metode *Equivalence Partioning* (EP). Metode *black box testing* merupakan salah satu metode pengujian perangkat lunak yang menguji aspek

fundamental sistem tanpa memperhatikan struktur logika perangkat lunak. EP berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. EP didasarkan pada premis masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas, menurut spesifikasi dari komponen tersebut, yang diperlakukan sama (*ekivalent*) oleh komponen tersebut. Pada pengujian ini harus diyakinkan bahwa masukan yang sama menghasilkan respon yang sama pula. Alasan menggunakan metode EP pada pengujian aplikasi KKN Mahasiswa Universitas Lampung ini adalah karena metode ini dapat digunakan untuk mencari kesalahan pada fungsi yang diberikan ke aplikasi dan dapat mengetahui kesalahan pada interface aplikasi sehingga dapat mengurangi masalah terhadap nilai masukan. Rancangan daftar pengujian dapat dilihat pada Tabel 3.1

Tabel 3.1 Rancangan Daftar Pengujian

No	Kelas Uji	Daftar Penguji	Skenario Uji	Hasil yang Diharapkan
1.	Versi Android	Pengujian kompatibilitas versi <i>operation system</i> android	Pengujian pada android versi 4.1 (<i>Jellybean</i>)	Kompatibel dengan android versi 4.1 (<i>Jellybean</i>)
			Pengujian pada android versi 4.4 (<i>Kitkat</i>)	Kompatibel dengan android versi 4.4 (<i>Kitkat</i>)
			Pengujian pada android versi 5.0 (<i>Lollipop</i>)	Kompatibel dengan android versi 5.0 (<i>Lollipop</i>)
2.	Ukuran layar	Pengujian Ukuran Layar pada android	Pengujian pada android dengan ukuran layar 4"	Tampilan terlihat baik dan sesuai pada android dengan ukuran layar 4"

Tabel 3.1 Rancangan Daftar Pengujian (lanjutan)

			Pengujian pada android dengan ukuran layar 5"	Tampilan terlihat baik dan sesuai pada android dengan ukuran layar 5"
			Pengujian pada android dengan ukuran layar 5.5"	Tampilan terlihat baik dan sesuai pada android dengan ukuran layar 5.5"
3.	<i>User Interface</i>	Pengujian pada Main Menu aplikasi KKN Universitas Lampung	Klik tombol menu "Pendaftaran KKN"	Menampilkan <i>layout</i> Pendaftaran KKN
			Klik tombol menu "Login"	Menampilkan <i>layout</i> Login
			Klik tombol menu "Petunjuk"	Menampilkan <i>layout</i> Petunjuk Aplikasi
			Klik tombol menu "About"	Menampilkan <i>layout</i> About Aplikasi
			Klik tombol "Jadwal Kegiatan KKN"	Menampilkan <i>layout</i> Jadwal Kegiatan KKN
		Pengujian pada Menu Mahasiswa aplikasi KKN Universitas Lampung	Klik tombol menu "Home"	Menampilkan <i>layout</i> Home Menu Mahasiswa
			Klik tombol menu "Info Penempatan"	Menampilkan <i>layout</i> Info Penempatan
			Klik tombol menu "Info DPL"	Menampilkan <i>layout</i> Info DPL
			Klik tombol menu "Lihat Nilai"	Menampilkan <i>layout</i> Lihat Nilai
			Klik tombol menu "Laporan Kegiatan"	Menampilkan <i>layout</i> Laporan Kegiatan
4.	Fungsi <i>Layout</i> Pendaftaran KKN	Pengujian pada <i>layout</i> Pendaftaran KKN	Klik tombol "Daftar"	Untuk mendaftar sebagai peserta KKN
			Klik tombol "Choose File"	Untuk memilih foto yang akan di-upload
5	Fungsi <i>Layout</i> Login	Pengujian pada <i>layout</i> Login	Klik tombol "Login"	Masuk ke Menu Mahasiswa bila NPM dan Token benar

Tabel 3.1 Rancangan Daftar Pengujian (lanjutan)

6	Fungsi <i>Layout</i> Petunjuk	Pengujian pada menu Petunjuk	Klik tombol menu “Petunjuk”	Menampilkan informasi petunjuk penggunaan aplikasi
7	Fungsi <i>Layout</i> <i>About</i>	Pengujian pada menu <i>About</i>	Klik tombol menu “ <i>About</i> ”	Menampilkan informasi mengenai aplikasi
8	Fungsi <i>Layout</i> Jadwal Kegiatan KKN	Pengujian pada Jadwal Kegiatan KKN	Klik tombol menu “Jadwal Kegiatan KKN”	Menampilkan <i>popup</i> jadwal kegiatan kkn
9	Fungsi <i>layout</i> <i>Home</i> Mahasiswa	Pengujian pada menu <i>Home</i> Menu Mahasiswa	Klik tombol menu “ <i>Home</i> ”	Menampilkan Home Menu Mahasiswa yang berisi jadwal kegiatan KKN
10	Fungsi <i>layout</i> Info Penempatan	Pengujian database Info Penempatan	Klik tombol menu “Info Penempatan”	Menampilkan informasi lokasi desa
11	Fungsi <i>layout</i> Info DPL	Pengujian database Info DPL	Klik tombol menu “Info DPL”	Menampilkan Info mengenai DPL dan KDPL
12	Fungsi <i>layout</i> Lihat Nilai	Pengujian database Nilai	Klik tombol menu “Lihat Nilai”	Menampilkan nilai yang didapat setelah pelaksanaan KKN selesai
13	Fungsi <i>layout</i> Laporan Kegiatan	Pengujian <i>database</i> laporan kegiatan	Klik tombol “Buat Laporan”	Untuk membuat laporan kegiatan KKN baru
			Klik tombol “Hapus Laporan”	Untuk menghapus laporan kegiatan

V. PENUTUP

5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan beberapa hal sebagai berikut :

1. Telah berhasil dibangun aplikasi KKN Mahasiswa Universitas Lampung yang dapat membantu mahasiswa dalam proses pendaftaran sebagai peserta KKN Universitas Lampung dan membantu mahasiswa untuk mendapatkan informasi mengenai KKN mulai dari jadwal pelaksanaan KKN, lokasi desa tempat mahasiswa ditempatkan, informasi mengenai Dosen Pembimbing Lapangan, melihat nilai yang didapat setelah KKN selesai, hingga pengiriman laporan kegiatan mahasiswa.
2. Berdasarkan hasil kuisisioner yang telah dibagikan kepada 5 orang responden untuk menguji *interface* dan fungsi dari aplikasi ini dapat disimpulkan bahwa *interface* dan fungsi dari aplikasi ini berjalan sesuai dengan yang diharapkan.
3. Berdasarkan hasil kuisisioner yang telah dibagikan kepada 50 mahasiswa, dari hasil perhitungan kuisisioner tersebut dapat disimpulkan bahwa aplikasi ini mendapatkan rata-rata persentase penilaian sebesar 87.12%. Berdasarkan persentase tersebut, aplikasi ini termasuk dalam kategori “Sangat Baik”.

5.2. Saran

Berdasarkan penelitian yang dilakukan maka diperoleh beberapa saran untuk pengembangan aplikasi ini lebih lanjut sebagai berikut :

1. Aplikasi dapat dikembangkan dengan menambahkan fitur yang dapat digunakan Dosen Pembimbing Lapangan mengelola laporan kegiatan yang telah dikirim mahasiswa melalui aplikasi ini.
2. Aplikasi dapat dikembangkan dengan memberi fitur *tracking* menuju lokasi desa untuk penelitian selanjutnya.
3. Diharapkan kegiatan KKN memiliki *project plan* dan *project implementation* sehingga kegiatan KKN dapat terdokumentasi dengan baik dan dapat dijadikan *open access* bagi seluruh *stakeholders* sebagai bahan rujukan untuk implementasi program-program pengembangan desa.

DAFTAR PUSTAKA

- Ayuliana. 2009. Testing dan Implementasi *Black Box Testing*. Jakarta
- Azwar, S. 2011. Sikap dan Perilaku. *Dalam: Sikap Manusia Teori dan Pengukurannya*. Yogyakarta: Pustaka Pelajar.
- BCS. 2001. *Standard For Software Component*. Inggris : *British Computer Society Specialist Interest Group in Software Testing*.
- BP-KKN. 2016. Buku Panduan Kuliah Kerja Nyata (KKN). Lampung: Universitas Lampung.
- Chasseur, dan Patel, Jm. 2013. *Enabling JSON Document Stores in Relational Systems*. *Sixteenth International Workshop on the Web and Databases*. WebDB 2013.
- Deviana, H. 2011. *Penerapan XML Web service* Pada Sistem Distribusi Barang. Yogyakarta: Universitas Gadjah Mada.
- Djarwanto, P.S. 1996. *Statistik Induktif*. BPFE-UGM. Yogyakarta.
- Fowler, Martin. 2004. *UML Distilled* Panduan Singkat Bahasa pemodelan Objek Standar, Edisi 3. Yogyakarta. Andi Publishing.
- Haviluddin. 2011. Memahami Penggunaan UML (*Unified Modelling Language*). Samarinda. Universitas Mulawarman.
- Hermawan, Rudi, Arief Hidayat, Victor Gayuh Utomo. 2015. Sistem Informasi Penjadwalan Kegiatan Belajar Mengajar Berbasis Web. Semarang. STMIK ProVisi Semarang.

- Ichwan, M, dan Fifin Hakiky. 2011. Pengukuran Kinerja Goodreads Application Programming Interface (Api) Pada Aplikasi Mobile Android. Bandung: Institut Teknologi Nasional Bandung.
- Jiang, F., Y. Lu. 2012. *Software testing model selection research based on yinyang testing theory*. In: *IEEE Proceeding of International Conference on Computer Science and Information Processing (CISP)*, pp. 590-594.
- Karuniawati, Suri, Sri Widowati, Iman Lukmanul Hakim. 2015. Implementasi Metode *Cause Effect Graphing* (CEG) dalam Pengujian Requirement Perangkat Lunak. Bandung : Unikom.
- Meildy, Bayu. 2014. Sistem Informasi Akademik Berbasis Web Di SMP Negeri 44 Bandung. Bandung : Unikom.
- Mustaqbal, M. Sidi, Roeri Fajri Firdaus, Hendra Rahmadi. 2015. Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. Bandung : Utama.
- Muwisnawangsa, Vandu R. 2017. Analisis Pengelompokan Mahasiswa Kkn Berdasarkan Kriteria Jenis Kelamin, Fakultas Dan Sekolah. Lampung : Universitas Lampung.
- Pratama, Danzen H. 2017. Pengembangan Sistem Pelaporan Kegiatan Kkn Berbasis Android. Lampung : Universitas Lampung.
- Pressman, Roger S. 2001. *Software Engineering A Practitioner's Approach Fifth Edition*. McGraw-Hill Companies, Inc, New York.
- Pressman, R.S. 2010. *Software Engineering: A Practitioner's Approach, 7th Edition*. New York: McGraw-HillCompanies, Inc.
- Safaat, N. 2011. Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung: Penerbit Informatika.