

**RANCANG BANGUN APLIKASI IDENTIFIKASI DAN KLASIFIKASI
KERUSAKAN RUMAH TINGGAL WARGA DAMPAK BENCANA
BERBASIS *MOBILE***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Afa Yura Putra

NIM : 115060800111025



PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

Rancang Bangun Aplikasi Identifikasi dan Klasifikasi Kerusakan Rumah
Tinggal Warga Dampak Bencana Berbasis *Mobile*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Afa Yura Putra
NIM: 115060800111025

Skripsi ini telah diuji dan dinyatakan lulus pada
15 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dr.Eng. Herman Tolle, S.T., M.T.
NIP: 19740823 200012 1 001

Rekyan Regasari MP, S.T., M.T.
NIK: 77041406120253

Mengetahui
Ketua Program Studi Informatika/Illmu Komputer

Drs. Marji, M.T
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 15 Januari 2016



Afa Yura Putra

NIM: 115060800111025



KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang selalu memberikan rahmat, hidayah dan cahaya-Nya sehingga penulis dapat menyelesaikan Proposal Skripsi dengan judul “Rancang Bangun Aplikasi Identifikasi dan Klasifikasi Kerusakan Rumah Tinggal Warga Dampak Bencana Berbasis *Mobile*”. Proposal skripsi ini diajukan sebagai salah satu syarat untuk mendapatkan gelar sarjana S-1 di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

Keberadaan skripsi ini tidak terlepas dari bimbingan dan bantuan dari berbagai pihak, untuk itu pada kesempatan ini penulis menyampaikan rasa terima kasih dan penghargaan sebesar- besarnya kepada :

1. Dr.Eng Herman Tolle, S.T., M.T. selaku Dosen Pembimbing I yang telah memberikan bimbingan, arahan dan motivasi hingga skripsi ini dapat terselesaikan
2. Rekyan Regasari Mardi Putri, S.T., M.T. selaku Dosen Pembimbing II yang telah memberikan pengetahuan, bimbingan dan arahan untuk kesempurnaan penulisan skripsi ini
3. Sutrisno, Ir., MT, selaku Ketua Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, dan Drs. Marji, MT., selaku Ketua Program Studi Teknik Informatika Universitas Brawijaya;
4. Kedua orang tua yang selalu memberikan doa dan semangat hingga telah banyak membantu lancarnya penulisan dalam skripsi ini
5. Seluruh dosen Program Studi Teknik Informatika Universitas Brawijaya, atas dukungan dan kerjasamanya;
6. Teman- teman Program Studi Teknik Informatika Universitas Brawijaya angkatan 2011 untuk bantuan dan semangatnya; dan
7. Semua pihak yang tidak dapat disebutkan satu- persatu.

Malang, 15 Januari 2016

Penulis
afayuraputra@gmail.com

ABSTRAK

Afa Yura Putra. 2015. : Rancang Bangun Aplikasi Identifikasi dan Klasifikasi Kerusakan Rumah Tinggal Warga Dampak Bencana Berbasis *Mobile*. Skripsi Program Studi Informatika/Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing: Dr. Eng.Herman Tolle, ST., MT. Dan Rekyan Regasari MP, ST., MT.

Di Indonesia, jika terjadi suatu bencana seperti tanah longsor, gunung meletus dan sebagainya, kerusakan yang ditimbulkan akan besar. Karena besarnya kerusakan yang timbul, pemerintah setempat seringkali kesulitan dalam melakukan pendataan kerusakan tersebut, sehingga diperlukan suatu sarana informasi yang cepat dan akurat mengenai apa yang dibutuhkan daerah tersebut. Untuk menunjang itu semua, perlunya disiapkan suatu metode ataupun teknologi yang dapat diaplikasikan menggunakan perangkat bergerak untuk keperluan pendataan, meliputi identifikasi kerusakan rumah warga di daerah bencana tersebut.

Sistem ini akan mengolah hasil masukan data identifikasi kerusakan bangunan. Yaitu dengan melakukan perhitungan menggunakan suatu rumus untuk mengetahui jumlah kebutuhan material. Aplikasi ini akan menghasilkan keluaran berupa jenis dan jumlah kebutuhan material. Sehingga proses untuk mengetahui kebutuhan material akan menjadi lebih cepat jika menggunakan aplikasi ini.

Dari hasil pengujian fungsional, aplikasi telah memenuhi kebutuhan fungsional yang ada. Lalu berdasarkan aspek pengujian usability yang meliputi aspek *learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction*, aplikasi memiliki rata-rata tingkat usability sebesar 81.23% yaitu dengan status sangat memuaskan. Berdasarkan hasil pengujian implementasi yang terdiri dari pengujian instalasi, akses data, dan penggunaan ruang penyimpanan, menunjukkan bahwa aplikasi dapat terpasang dan berjalan dengan baik serta mampu menangani akses data dari berbagai kondisi.

Kata Kunci : Bencana, Identifikasi, Kerusakan Rumah, Perangkat Bergerak

ABSTRACT

Afa Yura Putra. 2015. : Rancang Bangun Aplikasi Identifikasi dan Klasifikasi Kerusakan Rumah Tinggal Warga Dampak Bencana Berbasis *Mobile*. Skripsi Program Studi Informatika/Illmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing: Dr. Eng.Herman Tolle, ST., MT. Dan Rekyan Regasari MP, ST., MT.

In Indonesia, when a disaster happen, the damage will be huge. Because of the magnitude of the damage, local governments are often difficulties in collecting damaged data, so we need rapid and accurate information about what the area needed. To support it all, we need to prepared a method or technology which can be applied using mobile devices for data collection purposes, including the identification of damage homes in the disaster area.

This sistem will process the identification data input of building damage. By performing calculations using a formulation that has been determined. This application will produce the output of the type and amount of material needs. So that the process for determining material needs will be faster when using this application.

From the results of functional testing, the application has met the functional requirements. Then based aspects of usability testing that includes aspects of learnability, efficiency, memorability, errors, and satisfaction, the application has an average degree of usability of 81.23% with a status that is very satisfying. Based on test results of implementation that consists of installation testing, data access, and the use of storage space, indicating that the application can be installed and run well and be able to handle data access from a variety of conditions.

Keywords: *Disaster, Identifications, Damage Homes, Mobile Devices*

DAFTAR ISI

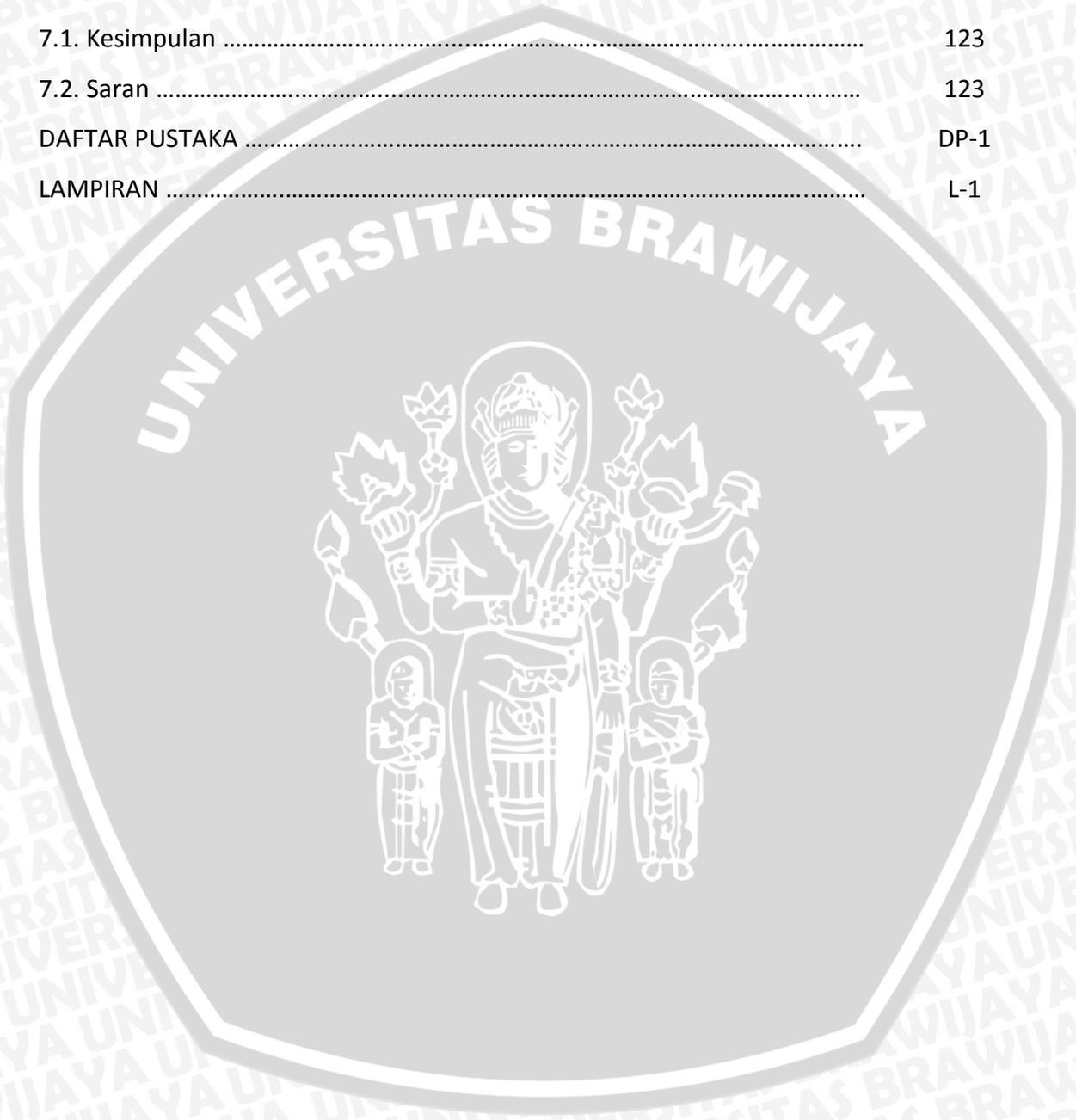
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR KODE	xv
DAFTAR PERSAMAAN	xvi
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
1.6. Metodologi Penulisan	3
BAB 2 TINJAUAN PUSTAKA	5
2.1. Kajian Pustaka	5
2.2. Dasar Teori	6
2.2.1. Rumah Tinggal	6
2.2.1.1. Pengertian Rumah Tinggal	6
2.2.1.2. Konsep Dasar Kebencanaan	6
2.2.1.3. Kerusakan Bangunan	7
2.2.1.4. Badan Nasional Penanggulangan Bencana (BNPB)	11
2.2.1.4.1. Tugas dan fungsi Badan Nasional Penanggulangan Bencana (BNPB)	12

2.2.5. Proses Bisnis Manual Identifikasi dan Klasifikasi Kerusakan Bangunan Dampak Bencana	12
2.2.6. Perangkat Bergerak	14
2.2.6.1. Aplikasi Perangkat Bergerak	15
BAB 3 METODOLOGI PENELITIAN	17
3.1. Studi Literatur	17
3.2. Analisa Kebutuhan Sistem	18
3.2.1. Kebutuhan Fungsional	18
3.3. Perancangan Sistem	19
3.3.1. Deskripsi Sistem	19
3.4. Implementasi Sistem	19
3.5. Pengujian Sistem	20
3.6. Pengambilan Kesimpulan dan Saran	20
BAB 4 PERANCANGAN	21
4.1. Analisis Kebutuhan.....	22
4.1.1. Gambaran Aplikasi	22
4.1.2. Identifikasi Aktor	23
4.1.3. Analisis Kebutuhan Fungsionalitas Sistem	23
4.1.3.1. <i>Use Case Diagram</i>	25
4.1.3.2. <i>Skenario Use Case</i>	26
4.1.4. Analisa Kebutuhan Non Fungsional	32
4.2. Perancangan Aplikasi Perangkat Bergerak	32
4.2.1. Perancangan <i>Activity Diagram</i>	33
4.2.2. Perancangan Arsitektur Sistem	39
4.2.2.1. Perancangan Pengolahan Data Identifikasi	40
4.2.3. Perancangan <i>Class Diagram</i>	42
4.2.4. Perancangan <i>Sequence Diagram</i>	46
4.2.5. Perancangan Basis Data	51
4.2.6. Perancangan Navigasi dan Antar Muka	52

4.2.6.1. Perancangan <i>Screen Flow</i>	52
4.2.6.2. Perancangan Antar Muka	54
BAB 5 IMPLEMENTASI	60
5.1. Spesifikasi Sistem	60
5.1.1. Spesifikasi Perangkat Keras	60
5.2. Batasan-Batasan implementasi	61
5.3. Implementasi Basis Data	62
5.4. Implementasi <i>Class</i> dan <i>Assets</i> pada File Program	67
5.5. Implementasi Kode Program	68
5.5.1. Implementasi Kode Program Perangkat Bergerak	68
5.5.2. Implementasi Kode Program <i>Web Service</i>	86
5.5.3. Implementasi Kode Program Untuk Perhitungan Data Kebutuhan Material.....	89
5.6. Implementasi Antarmuka	90
BAB 6 PENGUJIAN DAN ANALISIS	97
6.1. Pengujian Fungsional	98
6.2. Pengujian Non-Fungsional	102
6.2.1. Pengujian Usabilitas	102
6.2.2. Pengujian Kompatibilitas	104
6.3. Pengujian Implementasi	105
6.3.1. Pengujian Instalasi	105
6.3.2. Pengujian Akses Data	106
6.3.3. Pengujian Penggunaan Ruang Penyimpanan	108
6.4. Analisis	110
6.4.1. Analisis Pengujian Fungsional	110
6.4.2. Analisis Pengujian Non-Fungsional	112
6.4.2.1. Analisis Pengujian Usabilitas	112
6.4.2.2. Analisis Hasil Pengujian Kompatibilitas	117
6.4.3. Analisis Hasil Pengujian Implementasi	118



6.4.3.1. Analisis Pengujian Instalasi	118
6.4.3.2. Analisis Pengujian Akses Data	119
6.4.3.3. Analisis Hasil Pengujian Penggunaan ruang Penyimpanan	120
BAB 7 KESIMPULAN DAN SARAN	123
7.1. Kesimpulan	123
7.2. Saran	123
DAFTAR PUSTAKA	DP-1
LAMPIRAN	L-1



DAFTAR GAMBAR

Gambar 2.1. Diagram Alir Proses Bisnis	13
Gambar 3.1. Diagram Alir Proses	17
Gambar 4.1. Pohon Perancangan	21
Gambar 4.2. Gambaran Umum Aplikasi	23
Gambar 4.3. <i>Use Case Diagram</i>	25
Gambar 4.4. <i>Activity Diagram Login</i>	33
Gambar 4.5. <i>Activity Diagram Register</i>	34
Gambar 4.6. <i>Activity Diagram</i> Tambah Data Identifikasi	35
Gambar 4.7. <i>Activity Diagram</i> Hapus Data Identifikasi	36
Gambar 4.8. <i>Activity Diagram Edit</i> Data Identifikasi	37
Gambar 4.9. <i>Activity Diagram</i> Lihat Data Hasil Identifikasi Kerusakan Bangunan	38
Gambar 4.10. <i>Activity Diagram</i> Kirim Data Hasi Identifikasi Kerusakan Bangunan ke Server	39
Gambar 4.11. Arsitektur Sistem	40
Gambar 4.12. Rancangan Pengolahan Data Identifikasi	41
Gambar 4.13. Penjelasan Kelas pada Kelas <i>Diagram</i> Perangkat Bergerak (Bagian 1)	43
Gambar 4.14. Penjelasan Kelas pada Kelas <i>Diagram</i> Perangkat Bergerak (Bagian 2).....	43
Gambar 4.15. Penjelasan Kelas pada Kelas <i>Diagram</i> Perangkat Bergerak (Bagian 3).....	44
Gambar 4.16. Penjelasan Kelas pada Kelas <i>Diagram</i> Perangkat Bergerak (Bagian 4).....	45
Gambar 4.17. Penjelasan Kelas pada Kelas <i>Diagram</i> Perangkat Bergerak (Bagian 5).....	45
Gambar 4.18. <i>Sequence Diagram Login</i>	46
Gambar 4.19. <i>Sequence Diagram Register</i>	47

Gambar 4.20. <i>Sequence Diagram</i> Tambah Data	48
Gambar 4.21. <i>Sequence Diagram</i> Hapus Data Identifikasi.....	48
Gambar 4.22. <i>Sequence Diagram</i> Edit Data Identifikasi	49
Gambar 4.23. <i>Sequence Diagram</i> Lihat Data Identifikasi	50
Gambar 4.24. <i>Sequence Diagram</i> Kirim Data Hasil Identifikasi ke Server ..	50
Gambar 4.25. Perancangan Basis Data Identifikasi Kerusakan Bangunan..	51
Gambar 4.26. Peta Navigasi Aplikasi	52
Gambar 4.27. <i>Flow Screen</i> Aplikasi	53
Gambar 4.28. Rancangan Halaman Utama	54
Gambar 4.29. Rancangan Halaman <i>Login</i>	55
Gambar 4.30. Rancangan Halaman <i>Register</i>	56
Gambar 4.31. Rancangan Halaman <i>My Account</i>	57
Gambar 4.32. Rancangan Halaman Tambah Data	58
Gambar 4.33. Rancangan Halaman Lihat Data Hasil Identifikasi	59
Gambar 5.1. Hasil Implementasi Halaman <i>Login</i> dan <i>Register</i>	90
Gambar 5.2. Hasil Implementasi Halaman <i>My Account</i>	91
Gambar 5.3. Hasil Implementasi Halaman Identifikasi Bangunan	92
Gambar 5.4. Hasil Implementasi Halaman <i>View Data</i>	93
Gambar 5.5. Hasil Implementasi Halaman <i>Edit Data</i>	94
Gambar 5.6. Hasil Implementasi Halaman <i>Sync Data</i>	95
Gambar 5.7. Hasil Implementasi Halaman <i>View Data</i> pada <i>Server</i>	96
Gambar 6.1. Pohon Diagram Pengujian dan Analisis	97
Gambar 6.2. Aplikasi Terpasang Pada Perangkat Bergerak Android	118
Gambar 6.3. Pengujian Akses Data dari Berbagai Kondisi	120
Gambar 6.4. Grafik Ukuran Penggunaan Basis Data	121
Gambar 6.5. Grafik Ukuran Penggunaan Ruang yang Digunakan Aplikasi pada Media Penyimpanan Internal Perangkat Bergerak	121

DAFTAR TABEL

Tabel 2.1. Klasifikasi Tingkat Kerusakan Rumah Tinggal Warga	9
Tabel 2.2. Klasifikasi Tingkat Kerusakan Bangunan dan Kerusakan Atap ...	10
Tabel 2.3. Jenis dan Jumlah Penutup Atap Tiap m ²	10
Tabel 4.1. Identifikasi Aktor	23
Tabel 4.2. Daftar Kebutuhan Fungsional	24
Tabel 4.3. Skenario <i>Use Case Login</i>	26
Tabel 4.4. Skenario <i>Use Case Register</i>	27
Tabel 4.5. Skenario <i>Use Case</i> Tambah Data Identifikasi	27
Tabel 4.6. Skenario <i>Use Case</i> Hapus Data Identifikasi Kerusakan	28
Tabel 4.7. Skenario <i>Use Case</i> Edit Data Identifikasi Kerusakan	29
Tabel 4.8. Skenario <i>Use Case</i> Lihat Data Hasil Identifikasi	30
Tabel 4.9. Skenario <i>Use Case</i> Kirim Data Hasil Identifikasi ke Server	31
Tabel 4.10. Spesifikasi Kebutuhan Non Fungsional	32
Tabel 5.1. Spesifikasi Perangkat Keras Komputer	60
Tabel 5.2. Spesifikasi Perangkat Keras pada Perangkat Bergerak	60
Tabel 5.3. Spesifikasi Perangkat Lunak pada Komputer	61
Tabel 5.4. Spesifikasi Perangkat Lunak pada Perangkat Bergerak	61
Tabel 5.5. Struktur Tabel Identifikasi Kerusakan Bangunan di <i>Local Storage</i>	64
Tabel 5.6. Struktur Tabel Identitas Surveyor	65
Tabel 5.7. Struktur Tabel Lokasi <i>Survey</i>	65
Tabel 5.8. Struktur Tabel Identifikasi Kerusakan Bangunan di <i>Server</i>	65
Tabel 5.9. Berkas Kelas dan <i>Assets Layout</i> pada Aplikasi Perangkat Bergerak	67
Tabel 6.1. Kasus Uji Fungsional <i>Login</i>	98
Tabel 6.2. Kasus Uji Fungsional <i>Register</i>	98
Tabel 6.3. Kasus Uji Fungsional Tambah Data Identifikasi	99

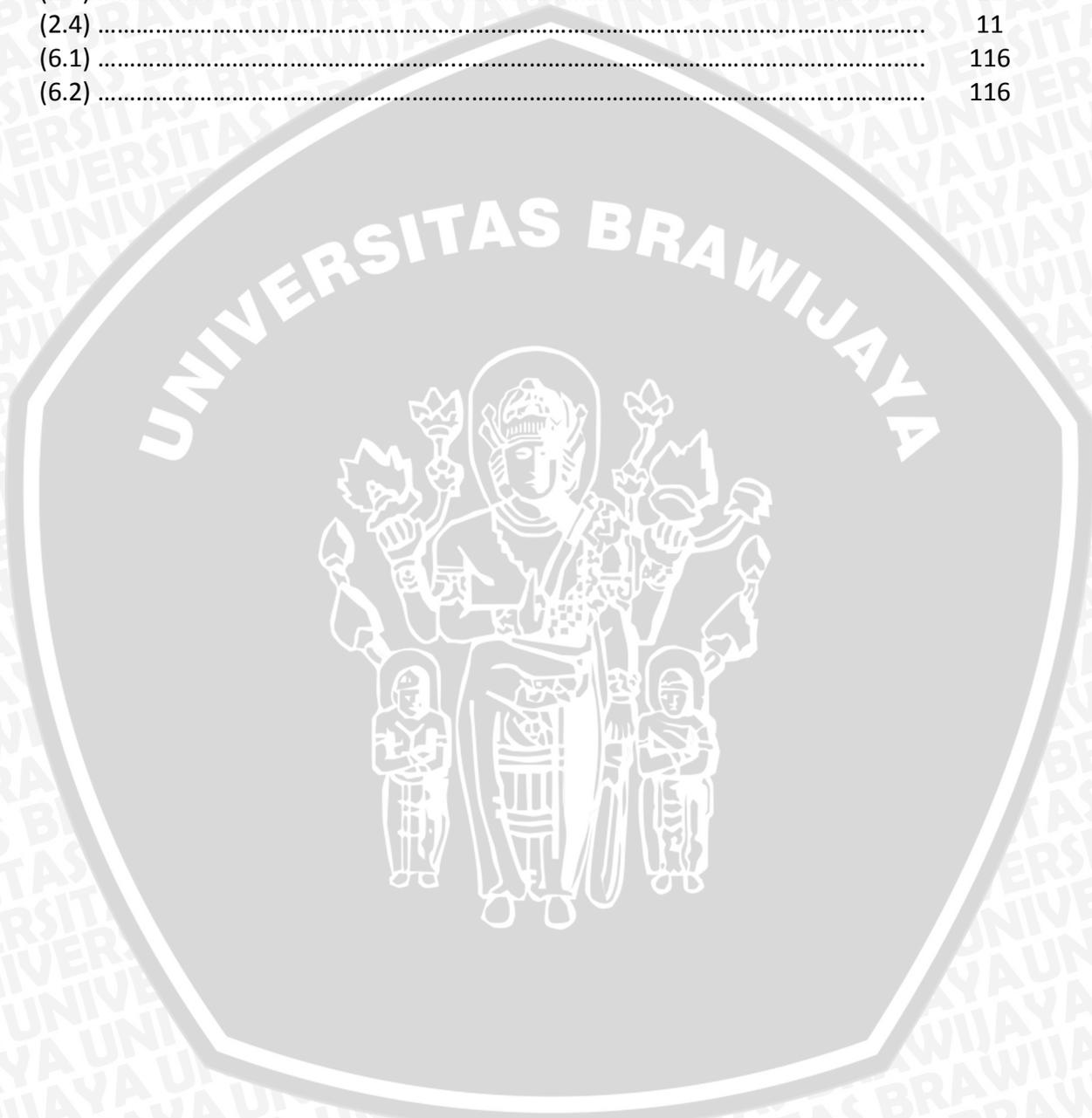
Tabel 6.4. Kasus Uji Fungsional Hapus Data Identifikasi	100
Tabel 6.5. Kasus Uji Fungsional Edit Data Identifikasi	100
Tabel 6.6. Kasus Uji Fungsional Lihat Data Hasil Identifikasi	101
Tabel 6.7. Kasus Uji Fungsional Kirim Data Hasil Identifikasi ke Server	101
Tabel 6.8. Kasus Uji Kompatibilitas Sistem Operasi Android 4.4	104
Tabel 6.9. Kasus Uji Kompatibilitas Sistem Operasi Android 5.1.1	105
Tabel 6.10. Hasil <i>Build Package</i> Pada Sistem Operasi Android	105
Tabel 6.11. Kasus Uji Instalasi pada Sitem Operasi Android	106
Tabel 6.12. Kasus Uji Akses Data dengan Internet <i>WiFi</i>	106
Tabel 6.13. Kasus Uji Akses Data dengan Internet Seluler	107
Tabel 6.14. Kasus Uji Akses Data Tanpa Koneksi Internet	108
Tabel 6.15. Kasus Uji Penggunaan Ruang Penyimpanan Aplikasi	108
Tabel 6.16. Kasus Uji Penggunaan Ruang Penyimpanan Basis Data	109
Tabel 6.17. Kasus Uji Penggunaan Ruang Aplikasi pada Penyimpanan Internal Perangkat Bergerak yang Terpakai	109
Tabel 6.18. Ukuran Basis Data yang Terpakai	109
Tabel 6.19. Ukuran Basis Data Setelah Data Terhapus	110
Tabel 6.20. Hasil Pengujian Fungsional	110
Tabel 6.21. Hasil Pengujian Usabilitas	112
Tabel 6.22. Presentase Skor Likert	114
Tabel 6.23. Indeks Presentase	114
Tabel 6.24. Hasil Status Pengujian Usabilitas	116
Tabel 6.25. Hasil Status Pengujian Kompatibilitas Android Versi 4.4	117
Tabel 6.26. Hasil Status Pengujian Kompatibilitas Android Versi 5.1	117
Tabel 6.27. Hasil Pengujian Instalasi	118
Tabel 6.28. Hasil Pengujian Akses Data dengan Internet <i>WiFi</i>	119
Tabel 6.29. Hasil Pengujian Akses Data dengan Internet Seluler	119
Tabel 6.30. Hasil Pengujian Akses Data Tanpa Koneksi Internet	119

DAFTAR KODE

Kode 5.1. Perintah SQLite Membuat Tabel data_identifikasi	63
Kode 5.2. Perintah SQLite Membuat Tabel dataidentifikasi	63
Kode 5.3. Perintah SQLite Membuat Tabel surveyor	63
Kode 5.4. Kode Program Kelas Identifikasi Bangunan.....	70
Kode 5.5. Kode Program Kelas <i>Login</i>	72
Kode 5.6. Kode Program Kelas <i>Register</i>	73
Kode 5.7 Kode Program Kelas <i>My Account</i>	75
Kode 5.8. Kode Program Kelas <i>FragmentViewData</i>	77
Kode 5.9. Kode Program Kelas <i>ViewData</i>	79
Kode 5.10. Kode Program Kelas <i>ViewAdapter</i>	80
Kode 5.11. Kode Program Kelas <i>EditData</i>	82
Kode 5.12. Kode Program Kelas <i>DBDataSource</i>	86
Kode 5.13. Kode Program Kelas <i>DB_Functions</i>	87
Kode 5.14. Kode Program insertidentifikasi.....	89
Kode 5.15. Kode Perhitungan Data Kebutuhan Material	89

DAFTAR PERSAMAAN

(2.1)	11
(2.2)	11
(2.3)	11
(2.4)	11
(6.1)	116
(6.2)	116



DAFTAR LAMPIRAN

Lampiran 1. Lembar Kuesioner Pengujian Usabilitas	L-1
Lampiran 2. Hasil Umpan Balik Kuesioner	L-3
Lampiran 3. Tabel Perhitungan Volume Kerusakan dan Kebutuhan Material	L-5



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Sebagai daerah dengan pertumbuhan ekonomi yang mulai meningkat, kebutuhan akan lahan kosong semakin bertambah. Ini berdampak pada pembukaan lahan baru yang sebelumnya merupakan hutan kini beralih fungsi sebagai pabrik, pusat perbelanjaan ataupun pemukiman penduduk sehingga memicu terjadinya bencana karna tanah tak mampu lagi menahan air dan tanah. Bencana yang diakibatkan hal ini adalah banjir dan tanah longsor. Bencana adalah Peristiwa atau rangkaian peristiwa yang mengancam dan mengganggu kehidupan dan penghidupan masyarakat yang disebabkan, baik oleh faktor alam dan/atau faktor non alam maupun faktor manusia sehingga mengakibatkan timbulnya korban jiwa manusia, kerusakan lingkungan, kerugian harta benda, dan dampak psikologis. (Definisi bencana menurut UU No. 24 tahun 2007). Secara geografis Indonesia merupakan negara kepulauan yang terletak pada pertemuan empat lempeng tektonik yaitu lempeng Benua Asia, Benua Australia, lempeng Samudera Hindia dan Samudera Pasifik. Pada bagian selatan dan timur Indonesia terdapat sabuk vulkanik (*volcanic arc*) yang memanjang dari Pulau Sumatera – Jawa - Nusa Tenggara – Sulawesi, yang sisinya berupa pegunungan vulkanik tua dan dataran rendah yang sebagian didominasi oleh rawa-rawa. Kondisi tersebut sangat berpotensi sekaligus rawan bencana seperti letusan gunung berapi, gempa bumi, tsunami, banjir dan tanah longsor. Data menunjukkan bahwa Indonesia merupakan salah satu negara yang memiliki tingkat kegempaan yang tinggi di dunia. (BNPB, 2011).

Masyarakat Indonesia, terutama yang memiliki tempat tinggal di daerah rawan bencana seperti pegunungan, kebanyakan merupakan masyarakat menengah kebawah yang rata – rata memiliki tingkat pengetahuan yang cenderung kurang. Seperti pengetahuan akan kebencanaan dan pengetahuan mengenai bagaimana membangun tempat tinggal yang baik dan benar. Mereka cenderung membangun tempat tinggalnya dengan seadanya, karena juga menyesuaikan dengan kondisi keuangan mereka.

Jadi seringkali jika terjadi suatu keadaan bencana seperti tanah longsor, gunung meletus dan sebagainya, kerusakan yang ditimbulkanpun akan semakin banyak.

Karena terlalu banyaknya kerusakan yang ditimbulkan tersebut, pemerintah setempat seringkali merasa kesulitan dalam melakukan pendataan akibat timbulnya kerusakan yang diakibatkan oleh terjadinya bencana alam tersebut. Karena pada awalnya sistem untuk melakukan pendataan yang digunakan masih menggunakan sistem manual, yaitu dengan mengisi data identifikasi kerusakan tersebut ke dalam suatu form berbentuk kertas, baru setelah itu data tersebut

harus dilakukan perhitungan secara manual satu persatu untuk menentukan jumlah kebutuhan material. Hal tersebutlah yang membuat pemerintah merasa kesulitan dalam melakukan pendataan kerusakan bangunan. Sehingga diperlukan suatu sarana informasi yang cepat dan akurat mengenai apa yang dibutuhkan daerah yang terkena bencana tersebut.

Teknologi Informasi, meliputi segala hal yang berkaitan dengan proses, penggunaan sebagai alat bantu, manipulasi, dan pengelolaan informasi. Sedangkan Teknologi Komunikasi merupakan segala hal yang berkaitan dengan penggunaan alat bantu untuk memproses dan mentransfer data dari perangkat yang satu menuju ke perangkat lainnya.

Perkembangan teknologi yang sangat cepat telah membawa manusia memasuki kehidupan yang berdampingan dengan informasi dan teknologi itu sendiri, yang berdampak pada sebagian orang untuk meninggalkan proses penelusuran informasi secara manual yang membutuhkan waktu lebih lama untuk mendapatkan atau menemukan informasi yang diinginkan.

Teknologi, jika dipadukan dengan suatu proses yang masih menggunakan sistem manual, akan dapat membuat proses yang terdapat pada sistem tersebut menjadi lebih cepat dari pada sebelumnya. Contohnya pada proses pendataan, jika proses pendataan yang masih menggunakan sistem manual digantikan dengan pendataan berbasis teknologi informasi, maka proses pendataan tersebut akan menjadi lebih cepat daripada sebelumnya. Apalagi dengan ditambahkan menggunakan perangkat media teknologi informasi saat ini seperti media perangkat bergerak atau *mobile*. Media tersebut dapat menggantikan penggunaan kertas yang dipakai pada sistem pendataan manual. Selain itu penggunaan perangkat bergerak dapat membuat sistem pendataan menjadi lebih praktis dan efisien, karena nantinya disini kita hanya tinggal memasukkan data ke dalam perangkat bergerak, lalu data tersebut akan langsung diolah oleh aplikasi yang terdapat di dalam perangkat bergerak.

Sehingga, dengan mengaplikasikan suatu metode berbasis teknologi pada proses pendataan, meliputi identifikasi dan klasifikasi kerusakan rumah warga di daerah bencana, penulis berharap dapat lebih mempermudah dan mempercepat kerja pemerintah dalam hal melakukan pendataan tersebut.

1.2 Rumusan Masalah

Berdasarkan paparan latar belakang tersebut, maka dapat dirumuskan masalah yang akan diselesaikan dalam penelitian ini yaitu

1. Bagaimana rancangan dan implementasi aplikasi perangkat bergerak yang sesuai untuk pendataan dampak akibat bencana khususnya pada daerah rawan ?
2. Bagaimana rancangan dan implementasi aplikasi yang dapat memproses data hasil identifikasi kerusakan yang terjadi pada rumah tinggal akibat bencana, sehingga menghasilkan suatu rekomendasi kebutuhan material berdasarkan data kerusakan.
3. Bagaimana performansi dari hasil implementasi dengan parameter usability, kompatibilitas, instalasi, akses data, dan penggunaan ruang penyimpanan.

1.3 Batasan Masalah

Untuk menghindari semakin melebarnya masalah, maka dari rumusan masalah yang telah dipaparkan, batasan masalah yang diterapkan adalah sebagai berikut :

1. Kriteria yang digunakan sebagai dasar penilaian diperoleh dari data kerusakan bangunan rumah tinggal.
2. Aplikasi ini dapat digunakan apabila bencana yang terjadi tidak sampai merusak sarana teknologi informasi.
3. Aplikasi ini hanya sebagai alat bantu bagi pihak surveyor untuk melakukan identifikasi dan klasifikasi kerusakan rumah tinggal. Namun keputusan akhir tetap berada di pihak surveyor.
4. Aplikasi ini akan menghasilkan suatu rekomendasi kebutuhan material untuk rumah tinggal berdasarkan data yang telah dimasukkan ke dalam aplikasi ini.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Membangun sistem perangkat bergerak yang dapat memudahkan pemerintah ataupun badan penanggulangan bencana untuk melakukan pendataan terhadap kerusakan yang diakibatkan terjadinya bencana alam.
2. Merancang sebuah sistem yang lebih cepat serta efisien untuk mempermudah proses identifikasi dan klasifikasi kerusakan rumah warga paska terjadinya bencana.

1.5 Manfaat Penelitian

Dengan dibangunnya aplikasi berbasis perangkat bergerak ini, penulis berharap dapat membantu pemerintah untuk lebih mempercepat dan mempermudah proses penanganan pasca bencana, terutama proses identifikasi dan klasifikasi kerusakan rumah warga yang timbul akibat terjadinya bencana tersebut. Sehingga nantinya kinerja pemerintah terutama dalam hal penanganan pasca bencana menjadi lebih akurat dan efisien dari sebelumnya.

1.6 Metodologi Penulisan

Sistematika penulisan yang diterapkan dalam tugas ini adalah sebagai berikut :

BAB 1 Pendahuluan

Pada bab ini dijabarkan latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB 2 Tinjauan Pustaka

Bab ini menjelaskan tentang teori-teori yang digunakan serta kajian pustaka yang mengenai teori identifikasi bencana, aplikasi mobile, web service, serta bahan kajian lain yang mendukung.

BAB 3 Metodologi Penelitian

Membahas tentang metode yang digunakan dalam penulisan yang terdiri dari studi literatur, studi lapangan, analisa kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem dan analisa hasil pengujian, serta pengambilan kesimpulan dan saran.

BAB 4 Analisis dan Perancangan

Membahas analisa tentang aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana berbasis mobile serta merancang sistem informasi yang sesuai dengan kebutuhan tersebut.

BAB 5 Implementasi

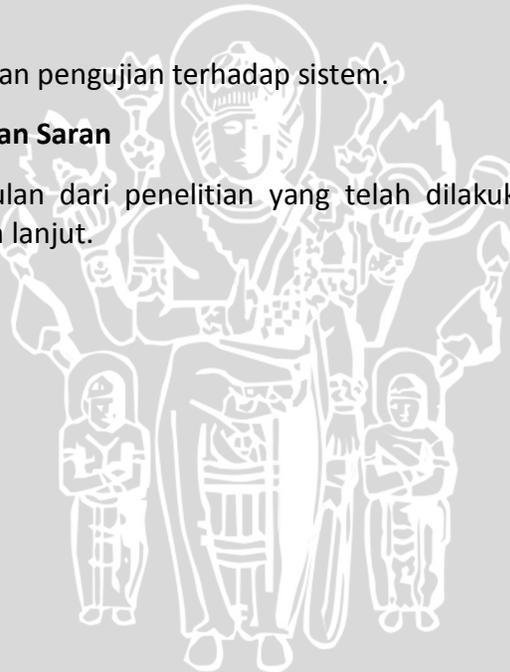
Membahas tentang hasil rancangan dari analisis kebutuhan serta implementasi dari hasil rancangan tersebut.

BAB 6 Pengujian

Memuat proses dan pengujian terhadap sistem.

BAB 7 Kesimpulan dan Saran

Memuat kesimpulan dari penelitian yang telah dilakukan serta saran untuk pengembangan lebih lanjut.



BAB 2 TINJAUAN PUSTAKA

Tinjauan Pustaka bertujuan untuk membentuk kerangka teori yang berasal dari literatur-literatur yang berhubungan dengan masalah yang diteliti. Bab ini membahas tinjauan pustaka yang digunakan untuk menunjang penulisan skripsi, yang mana di dalam tinjauan pustaka terdapat kajian pustaka yang membahas mengenai referensi dari tulisan sebelumnya yang terkait dengan penelitian ini, dan dasar teori yang membahas mengenai konsep dasar rumah tinggal, konsep dasar kerusakan bangunan, konsep dasar kebencanaan, dan konsep dasar aplikasi perangkat bergerak.

2.1 Kajian Pustaka

Di dalam kajian pustaka ini, akan dibahas mengenai penelitian sebelumnya yang terkait dengan penelitian yang dilakukan oleh penulis. Yang nantinya akan dapat digunakan sebagai referensi di dalam melakukan penelitian tersebut.

Dalam referensi tulisan sebelumnya yang berjudul "Aplikasi Sistem Informasi Manajemen Penanggulangan Pasca Bencana" (Kartadie, 2013), membahas mengenai suatu sistem informasi untuk melakukan manajemen dalam hal penanganan bencana alam. Fungsi dari aplikasi ini yaitu seperti Manajemen informasi data korban bencana, Manajemen *shelter* atau posko bencana, manajemen penyaluran logistic, manajemen bantuan tenaga relawan, dan manajemen informasi keadaan tiap lokasi penampungan.

Selain referensi tersebut, terdapat juga referensi lainya yang berjudul "Aplikasi Pelaporan Kejadian Bencana Secara Online Di Wilayah Aceh Berbasis SMS Gateway" (Nasaruddin, 2011), yang masih berkaitan dengan sistem kebencanaan. Aplikasi ini merupakan suatu aplikasi pelaporan adanya suatu kejadian bencana yang terjadi di wilayah aceh dengan menggunakan sistem online. Aplikasi ini menggunakan fasilitas SMS dan *google maps*. Proses pelaporan informasi dapat dibatasi secara individu, hanya member yang terdapat pada sistem saja yang dapat melakukan pelaporan. Yang mana hal tersebut digunakan untuk mempermudah proses validasi dan verifikasi pada sistem. Setelah proses verifikasi dan validasi, informasi yang diterima tersebut akan langsung dipetakan didalam peta wilayahh aceh uang terdapat pada *google maps*. Sehingga badan kebencanaan, maupun pemerintah bisa langsung mengetahui tentang kejadian bencana beserta lokasi terjadinya.

Berdasarkan referensi tersebut, penulis akan melakukan penelitian yang serupa yaitu mengenai penanganan kebencanaan, namun dengan sajian data yang berbeda, yaitu berupa klasifikasi dan identifikasi kerusakan bangunan rumah tinggal warga dampak setelah terjadinya suatu bencana. Selain itu,

perangkat yang akan digunakan penulis di dalam penelitian ini adalah perangkat bergerak yang dirasa lebih praktis dan efisien dibandingkan dengan perangkat desktop berbasis web seperti yang digunakan di dalam referensi tulisan pada penelitian sebelumnya.

2.2 Dasar Teori

2.2.1 Rumah Tinggal

2.2.1.1 Pengertian Rumah Tinggal

Dalam pengertian yang luas, rumah bukan hanya sebuah bangunan (struktural), melainkan juga tempat kediaman yang memenuhi syarat-syarat kehidupan yang layak, dipandang dari berbagai segi kehidupan masyarakat. Rumah dapat dimengerti sebagai tempat perlindungan, untuk menikmati kehidupan, beristirahat dan bersuka ria bersama keluarga. Di dalam rumah, penghuni memperoleh kesan pertama dari kehidupannya di dalam dunia ini. Rumah harus menjamin kepentingan keluarga, yaitu untuk tumbuh, memberi kemungkinan untuk hidup bergaul dengan tetangganya, dan lebih dari itu, rumah harus memberi ketenangan, kesenangan, kebahagiaan, dan kenyamanan pada segala peristiwa hidupnya. (Frick,2006:1). (Mega, 2013).

Rumah merupakan sebuah bangunan, tempat manusia tinggal dan melangsungkan kehidupannya. Disamping itu rumah juga merupakan tempat berlangsungnya proses sosialisasi pada saat seorang individu diperkenalkan kepada norma dan adat kebiasaan yang berlaku di dalam suatu masyarakat. Jadi setiap perumahan memiliki sistem nilai yang berlaku bagi warganya. Sistem nilai tersebut berbeda antara satu perumahan dengan perumahan yang lain, tergantung pada daerah ataupun keadaan masyarakat setempat. (Sarwono dalam Budihardjo, 1998 : 148). (Mega, 2013).

Rumah adalah bangunan yang berfungsi sebagai tempat tinggal atau hunian dan sarana pembinaan keluarga. (UU No.4 Tahun 1992 tentang Perumahan dan Permukiman).

Dari beberapa pengertian mengenai rumah yang telah dijelaskan sebelumnya dapat ditarik suatu kesimpulan bahwa rumah merupakan suatu bangunan yang dijadikan tempat tinggal selama jangka waktu tertentu, tempat bertumbuh, membina keluarga, makan, tidur,beraktivitas, dan lain sebagainya.

2.2.2. Konsep Dasar Kebencanaan

Kebencanaan merupakan suatu keilmuan yang membahas mengenai apa itu bencana, bagaimana menagani bencana, bagaimana melakukan pemulihan pasca bencana, dan lain sebagainya yang berhubungan dengan terjadinya bencana.

Di dalam suatu kebencanaan terdapat beberapa konsep dasar yaitu, (Hidayat, 2008) :

1. Kerentanan
Kerentanan adalah potensi untuk tertimpa kerusakan atau kerugian yang berkaitan dengan kapasitas untuk mengantisipasi suatu bahaya, mencegah bahaya dan memulihkan diri dari dampak bahaya.
2. Bencana
Bencana adalah berlangsungnya suatu kejadian bahaya yang luar biasa yang menimbulkan dampak pada komunitas-komunitas rentan dan mengakibatkan kerusakan, gangguan dan korban yang besar serta membuat kehidupan komunitas yang terkena dampak tidak berjalan dengan normal tanpa bantuan dari pihak luar.
3. Mitigasi
Mitigasi adalah segala bentuk langkah struktural (fisik) atau non struktural (misalnya, perencanaan penggunaan lahan, pendidikan publik) yang dilaksanakan untuk meminimalkan dampak merugikan dari kejadian-kejadian bahaya alam yang potensial timbul.
4. Kesiapsiagaan
Kesiapsiagaan adalah kegiatan-kegiatan dan langkah-langkah yang dilakukan sebelum terjadinya bahaya alam untuk meramalkan dan mengingatkan orang akan kemungkinan adanya kejadian bahaya tersebut, mengevakuasi orang dan harta benda jika mereka terancam dan untuk memastikan respon yang efektif (misalnya:dengan menumpuk bahan pangan).
5. Bantuan kemanusiaan, rehabilitasi dan rekonstruksi
Bantuan kemanusiaan, rehabilitasi dan rekonstruksi adalah segala bentuk kegiatan yang dilaksanakan setelah terjadinya bencana untuk menyelamatkan nyawa manusia dan memenuhi kebutuhan kemanusiaan yang mendesak, memulihkan kegiatan formal dan memulihkan infrastruktur fisik serta pelayanan masyarakat.

2.2.3. Kerusakan Bangunan

Menurut Peraturan Menteri Pekerjaan Umum No.24/PRT/M/2008 (2008:15) kerusakan bangunan adalah tidak berfungsinya bangunan atau komponen bangunan akibat penyusutan/berakhirnya umur bangunan, atau akibat ulah manusia atau perilaku alam seperti beban fungsi yang berlebih, kebakaran, gempa bumi, atau sebab lain yang sejenis. Intensitas kerusakan bangunan dapat digolongkan atas lima tingkat kerusakan (Idham, 2014), yaitu:

1. Kerusakan ringan non-struktural

Suatu bangunan dikategorikan mengalami kerusakan non-struktur apabila terjadi hal-hal sebagai berikut:

- a. Retak halus (lebar celah lebih kecil dari 0,075 cm) pada plesteran.
- b. Serpihan plesteran berjatuhan
- c. Mencakup luas yang terbatas.

2. Kerusakan ringan struktur

Suatu bangunan dikategorikan mengalami kerusakan struktur tingkat ringan apabila terjadi hal-hal sebagai berikut:

- a. Retak kecil (lebar celah lebih kecil dari 0,075 cm hingga 0,6 cm) pada dinding.
- b. Plester berjatuhan
- c. Mencakup luas yang besar
- d. Kerusakan bagian-bagian nonstruktur seperti cerobong, lisplang, dsb
- e. Kemampuan struktur untuk memikul beban tidak banyak berkurang
- f. Layak fungsi/huni

3. Kerusakan struktur tingkat sedang

Suatu bangunan dikategorikan mengalami kerusakan struktur tingkat sedang apabila terjadi hal-hal sebagai berikut:

- a. Retak besar (lebar celah lebih besar dari 0,6 cm) pada dinding
- b. Retak menyebar luas di banyak tempat, seperti pada dinding pemikul beban, kolom, cerobong miring, dan runtuh
- c. Kemampuan struktur untuk memikul beban sudah berkurang sebagian
- d. Layak fungsi/huni

4. Kerusakan berat

Suatu bangunan dikategorikan mengalami kerusakan struktur tingkat berat apabila terjadi hal-hal sebagai berikut:

- a. Dinding pemikul beban terbelah dan runtuh
- b. Bangunan terpisah akibat kegagalan unsur-unsur pengikat
- c. Kira-kira >50% elemen utama mengalami kerusakan
- d. Tidak layak fungsi/huni

5. Kerusakan Total

Suatu bangunan dikategorikan sebagai rusak total/roboh apabila terjadi hal-hal sebagai berikut:

- a. Bangunan rubuh seluruhnya (>65%)
- b. Sebagian besar komponen utama struktur rusak
- c. Tidak layak fungsi/huni

Tabel 2.1. Klasifikasi Tingkat Kerusakan Rumah Tinggal Warga

No.	Kategori dan Tingkat Kerusakan	
1	<ul style="list-style-type: none"> • Tingkat Kerusakan Atap: Berat • Tingkat Kerusakan Bangunan: Ringan 	
2	<ul style="list-style-type: none"> • Tingkat Kerusakan Atap: Berat • Tingkat Kerusakan Bangunan: Sedang 	
3	<ul style="list-style-type: none"> • Tingkat Kerusakan Atap: Berat • Tingkat Kerusakan Bangunan: Berat (Rumah runtuh) 	

Sumber : (Tim, 2014)

Secara lebih ringkas, klasifikasi tingkat kerusakan bangunan dan kerusakan atap dapat terlihat pada Tabel 2.2.

Tabel 2.2. Klasifikasi Tingkat Kerusakan Bangunan dan Kerusakan Atap

No.	Kategori	Tingkat Kerusakan		
		Ringan	Sedang	Berat
I	Kerusakan Bangunan	Hanya terdapat kerusakan pada bagian penutup atap (genteng) / elemen non struktural saja, tidak terdapat kerusakan struktural pada bangunan rumah tinggal	Terdapat kerusakan struktural < 50%	Terdapat kerusakan struktural > 50%
II	Kerusakan Atap	Kerusakan atap < 30%	Kerusakan atap antara 30% - 60%	Kerusakan atap > 60%

Sumber : (Tim, 2014)

Tabel 2.3. Jenis dan Jumlah Penutup Atap tiap m²

Jenis Penutup Atap	Jumlah Tiap m ²
Genteng Press	25 Buah
Karangpilang	25 Buah
Asbes	0.625 Buah

Sumber : (Panitia, 2002)

Klasifikasi kerusakan bangunan dan kerusakan atap yang tertulis pada Tabel 2.2, akan digunakan sebagai acuan dalam penilaian klasifikasi kerusakan bangunan pada penelitian ini. Dan untuk mengklasifikasikan

kerusakan bangunan dan kerusakan atap tersebut dilakukan secara *visual*, yaitu dengan melihat langsung kondisi kerusakan yang terjadi, lalu mengkategorikan kerusakan yang ada dengan data acuan tersebut.

Data acuan tersebut juga digunakan untuk melakukan perhitungan jumlah kebutuhan material berdasarkan data identifikasi yang nantinya akan dimasukkan ke dalam sistem pada aplikasi dalam penelitian ini. Yang mana rumus perhitungan yang digunakan tersebut dapat terlihat di Persamaan 2.1 sampai Persamaan 2.4. (Laboratorium, 2014)

$$Lb = p * l \dots\dots\dots (2.1)$$

$$La = 1.1 * Lb \dots\dots\dots (2.2)$$

$$Vr = La * \% (Persen)Kerusakan \dots\dots\dots (2.3)$$

$$Jm = Vr * Am \dots\dots\dots (2.4)$$

Dimana :

Lb = Luas bangunan

p = Panjang bangunan

l = Lebar bangunan

La = Luas permukaan atap bangunan

Vr = Volume kerusakan

Am = Jumlah kebutuhan material tiap m^2

Jm = Jumlah kebutuhan material keseluruhan

2.2.4. Badan Nasional Penanggulangan Bencana (BNPB)

Indonesia merupakan suatu negara yang memiliki gugusan pulau terbesar di dunia, yang man di pulau-pulau tersebut terdapat banyak sekali gunung api yang aktif hingga saat ini, yang mana gunung-gunung tersebut dapat sewaktu – waktu meletus. Berdasarkan hal tersebutlah Indonesia sering sekali terjadi suatu bencana, terutama bencana alam. Seringkali jika terjadi suatu bencana, pemerintah masih dianggap kurang tanggap dalam memberikan informasi mengenai bencana tersebut ataupun juga dalam melakukan penanganan akan terjadinya bencana tersebut. Maka dari itu pemerintah memutuskan untuk membentuk suatu badan pemerintah yang khusus menangani suatu kebencanaan, yang saat ini bernama BNPB.

Badan Nasional Penanggulangan Bencana (disingkat BNPB) adalah sebuah Lembaga Pemerintah Non Departemen yang mempunyai tugas membantu Presiden Republik Indonesia dalam: mengkoordinasikan perencanaan dan pelaksanaan kegiatan penanganan bencana dan kedaruratan secara terpadu, serta melaksanakan penanganan bencana dan kedaruratan mulai dari sebelum, pada saat, dan setelah terjadi bencana yang meliputi pencegahan, kesiapsiagaan, penanganan darurat, dan pemulihan.

BNPB dibentuk berdasarkan Peraturan Presiden Nomor 8 Tahun 2008. Sebelumnya badan ini bernama Badan Koordinasi Nasional Penanggulangan Bencana yang dibentuk berdasarkan Peraturan Presiden Nomor 83 Tahun

2005, menggantikan Badan Koordinasi Nasional Penanggulangan Bencana dan Penanganan Pengungsi yang dibentuk dengan Keputusan Presiden Nomor 3 Tahun 2001. (BNPB, 2011)

2.2.4.1. Tugas dan Fungsi Badan Nasional Penanggulangan Bencana (BNPB)

Dalam menjalankan tugasnya, BNPB selaku badan khusus yang dibentuk oleh pemerintah memiliki tugas-tugas dan fungsi utama. Tugas utama BNPB sebagai badan penanggulangan bencana adalah (BNPB, 2011) :

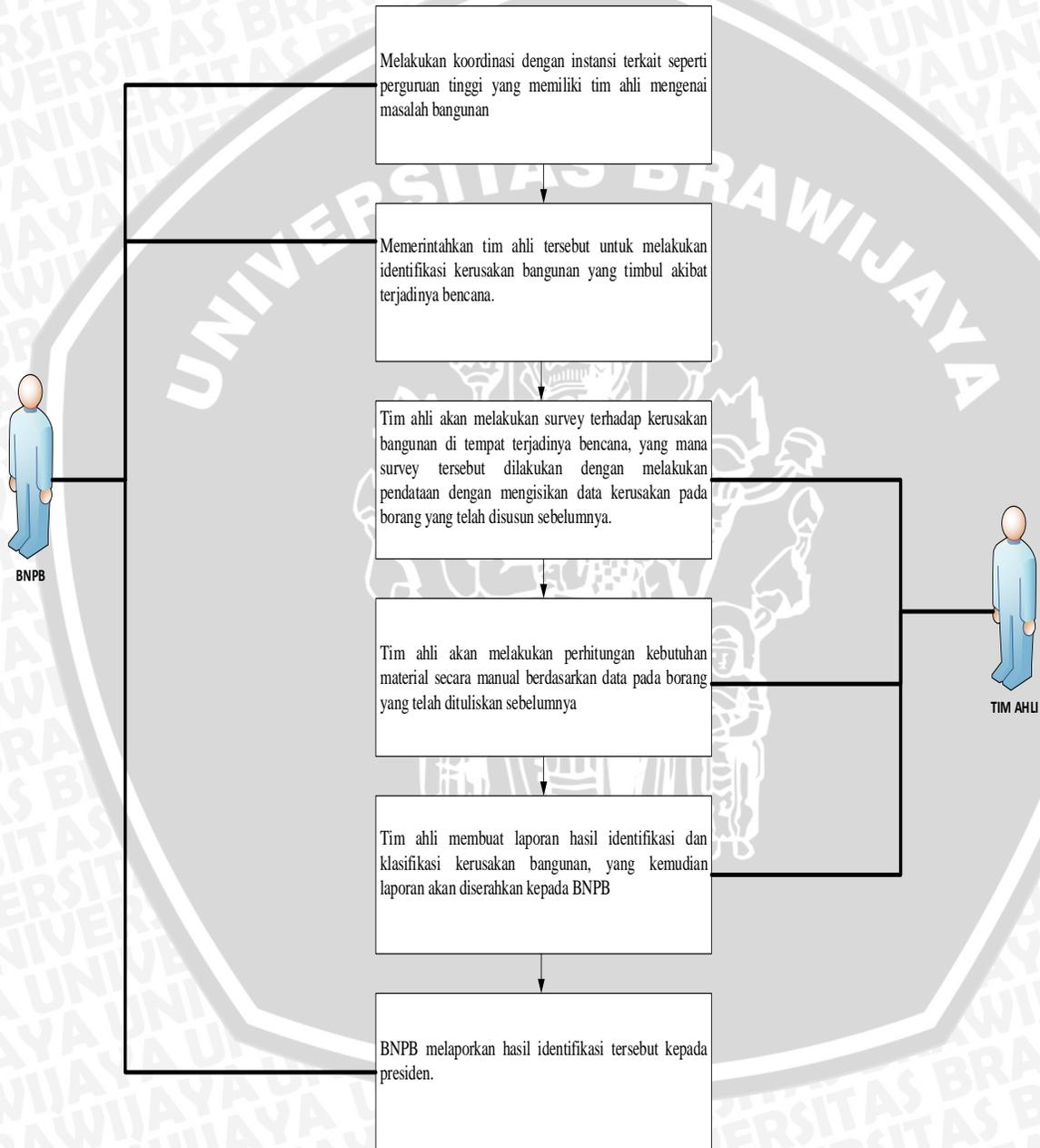
1. Memberikan pedoman dan pengarahan terhadap usaha penanggulangan bencana yang mencakup pencegahan bencana, penanganan tanggap darurat, rehabilitasi, dan rekonstruksi secara adil dan setara
2. Menetapkan standardisasi dan kebutuhan penyelenggaraan penanggulangan bencana berdasarkan peraturan perundang-undangan
3. Menyampaikan informasi kegiatan penanggulangan bencana kepada masyarakat
4. Melaporkan penyelenggaraan penanggulangan bencana kepada Presiden setiap sebulan sekali dalam kondisi normal dan setiap saat dalam kondisi darurat bencana
5. Menggunakan dan mempertanggungjawabkan sumbangan / bantuan nasional dan internasional
6. Mempertanggungjawabkan penggunaan anggaran yang diterima dari Anggaran Pendapatan dan Belanja Negara
7. Melaksanakan kewajiban lain sesuai dengan peraturan perundang-undangan
8. Menyusun pedoman pembentukan Badan Penanggulangan Bencana Daerah.

Selain memiliki tugas – tugas pokok seperti yang telah disebutkan sebelumnya, BNPB juga memiliki fungsi utama dalam menjalankan tugasnya, yaitu melakukan perumusan dan penetapan kebijakan penanggulangan bencana dan penanganan pengungsi dengan bertindak cepat dan tepat serta efektif dan efisien.

2.2.5. Proses Bisnis Manual Identifikasi dan Klasifikasi Kerusakan Bangunan Dampak Bencana

Dalam menangani kerusakan bangunan pada daerah pasca terjadinya bencana, BNPB akan melakukan koordinasi dengan instansi terkait untuk melakukan proses identifikasi dan klasifikasi kerusakan bangunan pasca terjadinya suatu bencana, dalam hal ini BNPB bekerjasama dengan tim ahli bangunan yang biasanya tim ahli tersebut ditunjuk dari perguruan tinggi .

Yang mana proses bisnis manual dalam melakukan identifikasi dan klasifikasi kerusakan bangunan pasca terjadinya suatu bencana terlihat di dalam Gambar 2.1.



Gambar 2.1 Diagram Alir Proses Bisnis Manual

Dari diagram alir pada Gambar 2.1 dapat terlihat bahwa proses yang dibutuhkan untuk melakukan survey identifikasi kerusakan bangunan masih dirasa kurang efisien karena masih ada proses manual yang dikerjakan untuk melakukan pendataan kerusakan bangunan dan proses perhitungan kebutuhannya, maka dari itulah, untuk lebih mempercepat proses tersebut, maka dari itu dilakukanlah penelitian ini, yaitu dengan membuat suatu aplikasi identifikasi kerusakan bangunan, yang mana dengan adanya aplikasi ini, proses pendataan kerusakan bangunan bisa menjadi lebih cepat, serta proses perhitungan kebutuhan akan secara otomatis dilakukan oleh sistem di dalam aplikasi, sehingga, keseluruhan proses identifikasi bangunan bisa menjadi lebih cepat dan efisien daripada sebelumnya.

2.2.6. Perangkat Bergerak

Perkembangan akan teknologi dari tahun ke tahun memperlihatkan suatu peningkatan yang signifikan. Yang mana saat ini teknologi berkembang menjadi suatu perangkat perangkat yang lebih ringkas ataupun lebih kecil daripada sebelumnya, namun memiliki fungsi yang sama, bahkan bisa lebih canggih daripada sebelumnya. Biasanya perangkat tersebut disebut sebagai perangkat bergerak, karena kita bisa membawa dan menggunakan perangkat tersebut dimanapun dan kemanapun yang kita inginkan.

Perangkat bergerak atau dalam bahasa inggris disebut dengan *mobile device* adalah perangkat kecil yang memiliki kemampuan komputasi terbatas. Perangkat bergerak sering juga disebut sebagai perangkat genggam atau komputer genggam. Perangkat bergerak pada umumnya terdiri dari dua bagian, yakni layar *display* sebagai perangkat keluaran dan keyboard atau layar sentuh sebagai masukan. Khusus untuk perangkat telepon pintar (*Smartphone*), umumnya perangkat layar visual selain berfungsi sebagai perangkat keluaran, juga berfungsi sebagai perangkat masukan, karena sifatnya yang memiliki layar sentuh pada keseluruhan tampilannya. (Zaki, 2008)

Karena fungsinya yang bervariasi, maka sebuah organisasi T38 dan DuPont *Global Mobility* merumuskan definisi standar dari perangkat bergerak (Zaki, 2008), yaitu :

1. *Limited Data Mobile Device* : Perangkat yang ukuran layarnya kecil, umumnya layar hanya menampilkan teks, layanan data hanya terbatas ke SMS dan WAP. Contoh perangkat ini adalah telepon seluler.
2. *Basic Data Mobile Device* : Perangkat yang ukuran layarnya menengah. Memiliki navigasi menggunakan menu atau ikon. Layanan yang ditamarkan antara lain surel (*email*), daftar alamat (Kontak), SMS, dan *Web Browser*. Contoh perangkat ini adalah telepon pintar (*Smartphone*).

3. *Enhanced Data Mobile Device* : Perangkat yang ukuran layarnya besar, biasanya menggunakan pena *stylus* untuk layar sentuhnya, dan memiliki fitur – fitur layanan dasar yang dimiliki oleh piranti sebelumnya ditambah adanya kemampuan untuk menambah berbagai aplikasi seperti *Microsoft Office*, dan portal internet. Contoh perangkat ini adalah *Pocket PC*, dan *Tablet*.

2.2.6.1. Aplikasi Perangkat Bergerak

Seiring dengan semakin berkembangnya suatu teknologi dalam perangkat bergerak, maka aplikasi yang dijalankan di dalam perangkat bergerak tersebut akan semakin berkembang pula, karena aplikasi perangkat bergerak yang dibuat haruslah menyesuaikan dengan teknologi apa yang terdapat di dalam perangkat bergerak tersebut.

Aplikasi perangkat bergerak merupakan aplikasi yang dapat dijalankan walaupun pengguna berpindah atau karena pengguna berpindah. Pemrograman aplikasi bergerak tidak banyak berbeda dengan pemrograman aplikasi bergerak tidak banyak berbeda dengan pemrograman konvensional pada *PC*. Aspek karakteristik dari perangkat bergerak sering mempengaruhi arsitektur dan implementasi dari aplikasi tersebut. Dalam pemrograman aplikasi bergerak berbagai aspek teknis perangkat lebih menonjol karena memiliki banyak keterbatasan dibandingkan computer konvensional atau *PC*. (Budi Darytamo dkk, 2007:1). (Fawzila, 2012)

Perangkat bergerak (*mobile*) memiliki banyak jenis dalam hal ukuran, desain dan layout, tetapi memiliki kesamaan karakteristik yang sangat berbeda dari sitem desktop. (Fawzila, 2012) :

1. Ukuran yang kecil
Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. Memory yang kecil
Perangkat *mobile* juga memiliki *memory* yang kecil, yaitu *primary* (*RAM*) dan *secondary* (*disk*). Pembatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan jumlah dari *memory*, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.
3. Daya proses yang terbatas
Sistem *mobile* tidaklah setangguh rekan mereka yaitu desktop. Ukuran, teknologi dan biaya adalah beberapa faktor yang mempengaruhi status dari sumber daya ini. Seperti *harddisk* dan *RAM*, anda dapat menemukan mereka dalam ukuran yang pas dengan sebuah kemasan kecil.

4. Mengonsumsi daya yang rendah

Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin *desktop*. Perangkat ini harus menghemat daya karena mereka berjalan pada keadaan dimana daya yang disediakan dibatasi oleh baterai-baterai.

5. Kuat dan dapat diandalkan

Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

6. Koneksi yang terbatas

Perangkat *mobile* memiliki *bandwith* rendah, beberapa dari mereka bahkan tidak tersambung. Kebanyakan dari mereka menggunakan koneksi *wireless*.

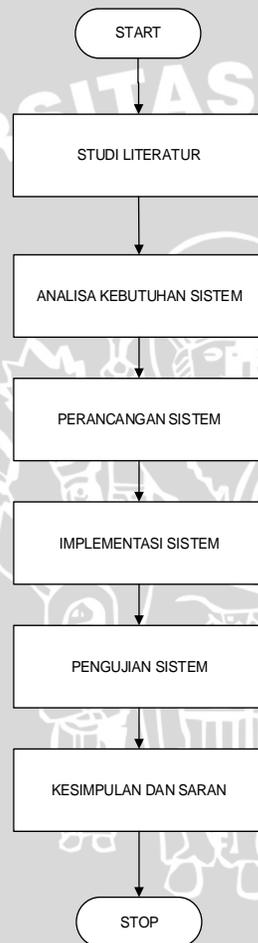
7. Masa hidup yang pendek

Perangkat bergerak memiliki ukuran yang tidak sebesar perangkat komputer, sehingga penyedia dayanya pun tidak sebesar di komputer, maka dari itu daya yang terdapat pada perangkat bergerak tidak akan dapat bertahan lama dibandingkan dengan daya pada komputer.



BAB 3 METODOLOGI PENELITIAN

Dalam bab ini akan dibahas metodologi yang digunakan dalam skripsi ini, meliputi : studi literatur, analisa kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian sistem, serta kesimpulan dan saran. Berikut diagram alir yang digunakan dalam metodologi ini.



Gambar 3.1 Diagram Alir Proses

3.1. Studi Literatur

Studi literatur dilakukan bertujuan untuk mempelajari serta memahami konsep-konsep sistem agar ketika dilakukan perancangan tidak terlalu mengalami kendala. Pada tahap studi literatur ini mempelajari mengenai

teori-teori yang digunakan dalam pengerjaan skripsi. Teori-teori untuk pendukung penulisan skripsi diperoleh dari jurnal, buku, e-book, dan penelitian sebelumnya yang topic pembahasannya sama atau berhubungan dengan skripsi. Hal utama yang dibahas dalam pengerjaan skripsi ini adalah aplikasi *mobile*, sistem penanganan pasca bencana, serta identifikasi dan klasifikasi kerusakan rumah warga dampak bencana.

3.2. Analisa Kebutuhan Sistem

Analisa kebutuhan system dilakukan untuk mengidentifikasi kebutuhan pada sistem aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana ini, ataupun juga untuk menentukan apa saja yang dapat dilakukan oleh sistem. Kebutuhan yang dimaksud adalah kriteria rekomendasi, bobot kriteria, kebutuhan user dari segi administrator maupun surveyor. Pada tahap ini dijelaskan mengenai batasan sistem serta tujuan yang dapat dicapai oleh pengguna. Pada analisa kebutuhan digunakan *use case diagram* akan digunakan untuk menjelaskan interaksi antara aktor dengan sistem tersebut serta *activity diagram* digunakan untuk menjelaskan proses pada interaksi yang terdapat di dalam *use case diagram*.

3.2.1. Kebutuhan Fungsional

Kegiatan analisis kebutuhan perangkat lunak meliputi analisis spesifikasi perangkat lunak. Metode analisis menggunakan bahasa pemodelan UML (*Unified Modeling Language*). *Use Case Diagram* digunakan untuk mendeskripsikan kebutuhan-kebutuhan dan fungsionalitas sistem dari perspektif *user*. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan (*requirements*) sistem yang kemudian akan dimodelkan dalam diagram *use case*.

Kebutuhan fungsional yang nantinya akan disediakan oleh aplikasi ini adalah :

- Aplikasi mampu untuk menampilkan halaman untuk melakukan *login* yang digunakan pengguna untuk masuk ke dalam sistem dan halaman *register* yang digunakan pengguna untuk melkaukan registrasi ke dalam sistem.
- Aplikasi mampu menampilkan halaman untuk input data identitas bangunan yang akan dilakukan survey, yaitu meliputi lokasi, jenis bangunan, fungsi bangunan, jumlah lantai, identitas pemilik bangunan, perkiraan luas bangunan, serta tingkat kerusakan bangunan, untuk input data identifikasi kerusakan pada bangunan rumah tinggal yang meliputi penutup atap, dan kelengkapannya, serta untuk input keterangan tambahan yang meliputi kondisi air bersih dan keterangan lainnya.

- Aplikasi mampu menampilkan data rekomendasi jenis kebutuhan beserta jumlahnya berdasarkan masukan data identifikasi kerusakan bangunan

3.3. Perancangan Sistem

Perancangan arsitektur sistem adalah tahap dimana penulis mulai merancang suatu sistem yang mampu memenuhi semua kebutuhan fungsional aplikasi dalam tugas akhir ini. Teori-teori dari pustaka dan data dari sample digabungkan dengan ilmu yang didapat diimplementasikan untuk merancang serta mengembangkan suatu sistem identifikasi dan klasifikasi kerusakan rumah warga dampak bencana.

3.3.1. Deskripsi Sistem

Sistem yang akan dibuat ini akan mengolah hasil masukan data identifikasi kerusakan bangunan. Yaitu dengan melakukan perhitungan menggunakan Persamaan 2.1 sampai Persamaan 2.4 yang terdapat pada bab sebelumnya. Sehingga nantinya pada sistem ini akan dihasilkan keluaran berupa jenis kebutuhan material beserta jumlahnya.

Pada tahap awal, *surveyor* akan memasukkan data identitas bangunan yang akan dilakukan identifikasi. Lalu *surveyor* akan melakukan identifikasi terhadap bangunan tersebut, yaitu dengan melakukan pengukuran luas kerusakan, dan identifikasi jenis material yang terdapat pada bangunan tersebut. Setelah itu *surveyor* akan memasukkan data hasil identifikasi kerusakan tersebut ke dalam sistem. Dari hasil identifikasi kerusakan tersebut akan diolah sedemikian rupa, sehingga nantinya sistem dapat menentukan jenis kebutuhan material beserta jumlahnya.

3.4. Implementasi Sistem

Implementasi aplikasi dilakukan berdasarkan perancangan aplikasi. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman perangkat bergerak yaitu menggunakan implementasi basis data SQLite dan MySQL dengan aplikasi XAMPP dan bahasa pemrograman perangkat bergerak sistem operasi *android* (Java SDK, XML). Implementasi aplikasi meliputi :

1. Pembuatan antar muka
2. Melakukan perhitungan luas kerusakan bangunan.
3. Melakukan perhitungan untuk penentuan jumlah kebutuhan material.
4. Menghasilkan *output* berupa jenis dan jumlah kebutuhan material

3.5. Pengujian Sistem

Pengujian dilakukan agar dapat menunjukkan bahwa perangkat lunak mampu bekerja sesuai dengan spesifikasi dan kebutuhan penggunanya dan mengetahui kinerja dan performa aplikasi. Pengujian sistem dilakukan menggunakan teknik *blackbox*. Pengujian *blackbox* digunakan untuk menguji fungsionalitas dari sistem. Pada pengujian validasi (*validation testing*) menggunakan teknik *blackbox testing*, pengujian *usability* menggunakan metode kuisiner. Pada kuisiner ini, responden akan memberikan nilai atau skor 1 sampai 5. Kuisiner ini nantinya akan disebarkan secara langsung dengan cara mendatangi responden, yaitu beberapa orang surveyor yang dahulunya pernah menggunakan sistem manual untuk melakukan identifikasi dan klasifikasi kerusakan bangunan, selain itu responden juga harus pernah menggunakan perangkat bergerak dalam kehidupan sehari-harinya. Kriteria khusus pada responden seperti usia dan jenis kelamin diabaikan. Parameter yang digunakan pada pengujian ini yaitu :

1. Kemudahan dalam menyelesaikan tugas dasar (*Learnability*)
2. Kecepatan dalam menyelesaikan tugas (*Efficiency*)
3. Kemudahan dalam mengingat penggunaan aplikasi (*Memorability*)
4. Kemungkinan terjadi kesalahan dan cara mengatasinya (*Errors*)
5. Tingkat kepuasan pengguna (*Satisfaction*)

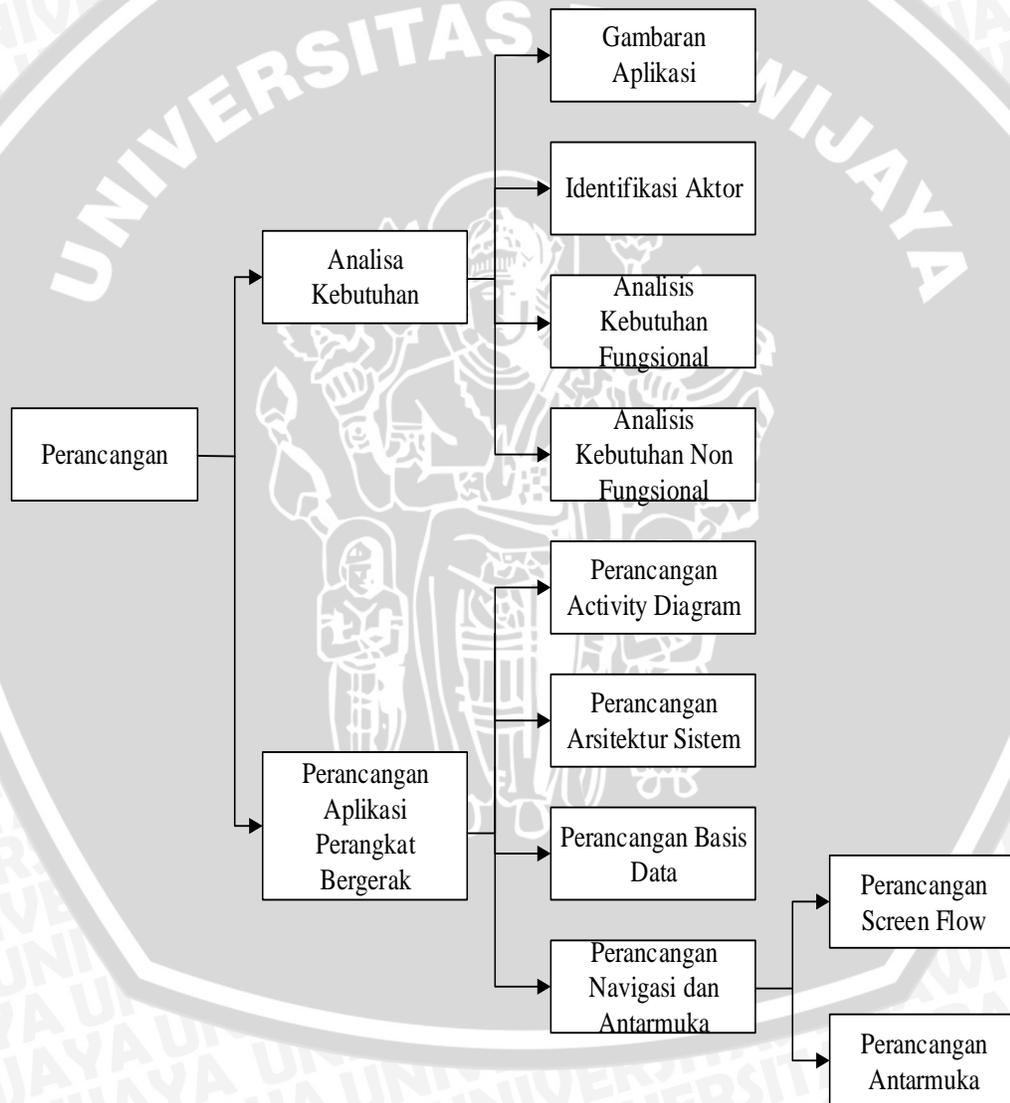
Langkah selanjutnya setelah melakukan pengujian, yaitu melakukan analisa terhadap hasil pengujian sehingga didapatkan kesimpulan dari pengembangan perangkat lunak yang telah dilakukan. Apabila pengujian menghasilkan data yang sesuai dengan kriteria dan nilai yang baik dari para responden, maka sistem dapat dikatakan sesuai dengan tujuan.

3.6. Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan dan didasarkan pada kesesuaian antara teori dan praktik. Kesimpulan diambil untuk menjawab rumusan masalah yang sudah dijabarkan sebelumnya. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dalam pembuatan Aplikasi Identifikasi dan Klasifikasi Kerusakan Rumah Tinggal Warga Dampak Bencana Berbasis Mobile dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

BAB 4 PERANCANGAN

Pada bab ini penulis akan membahas tentang perancangan sistem dari aplikasi perangkat bergerak identifikasi dan klasifikasi kerusakan rumah warga dampak bencana. Perancangan sistem ini terdiri dari analisis kebutuhan perangkat lunak dan perancangan perangkat lunak. Gambar 4.1 merupakan struktur dan langkah – langkah rancangan aplikasi Identifikasi dan Klasifikasi Kerusakan Rumah Warga Dampak Bencana Berbasis *Mobile*..



Gambar 4.1. Pohon Perancangan

4.1. Analisis Kebutuhan

Tahap analisis kebutuhan ini akan melakukan identifikasi semua kebutuhan aplikasi perangkat identifikasi dan klasifikasi kerusakan rumah warga dampak bencana. Terdapat beberapa tahap dalam analisa kebutuhan ini antara lain gambaran aplikasi, identifikasi aktor, daftar kebutuhan fungsional dan non fungsional dengan menggunakan use case diagram. Analisa kebutuhan ini bertujuan untuk memberikan analisa secara jelas tentang apa-apa saja kebutuhan yang disediakan oleh sistem demi memenuhi kebutuhan pengguna.

4.1.1. Gambaran Aplikasi

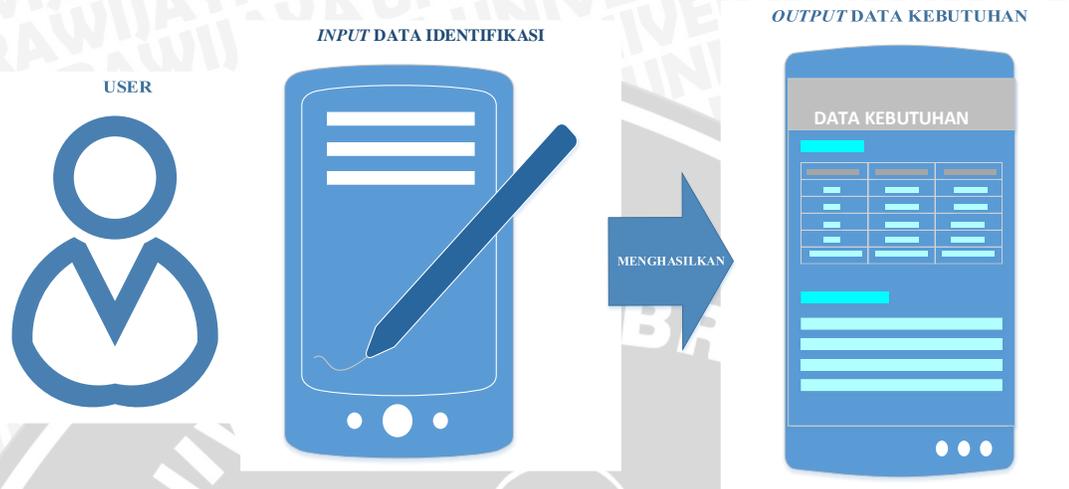
Aplikasi perangkat bergerak identifikasi dan klasifikasi kerusakan rumah warga dampak bencana ini merupakan aplikasi yang bertujuan untuk membantu kerja badan penanggulangan bencana untuk melakukan *survey* terhadap kerusakan-kerusakan yang terjadi terhadap rumah tinggal warga pasca terjadinya suatu bencana, yang mana untuk sebelumnya *survey* yang dilakukan masih menggunakan cara manual, yaitu dengan mengisikan data kerusakan pada form kertas, lalu melakukan perhitungan secara manual berdasarkan form – form yang telah diisi tersebut.

Pengguna aplikasi dalam hal ini *surveyor*, dapat melakukan *survey* dengan langsung mengisikan data kerusakan pada aplikasi ini. Yang mana data kerusakan yang dimasukkan meliputi identitas bangunan yaitu lokasi bangunan, jenis bangunan, fungsi bangunan, jumlah lantai, identitas pemilik bangunan, perkiraan luas bangunan, dan tingkat kerusakan bangunan. Data identifikasi kerusakan komponen struktural meliputi kerusakan penutup atap dan kelengkapan atap. Dan yang terakhir merupakan data keterangan tambahan. Selain memasukkan data tersebut, *surveyor* juga dapat menambahkan data berupa foto terhadap bangunan yang dilakukan *survey* tersebut.

Untuk membedakan data rumah satu dengan lainnya yang telah dilakukan *survey*, maka aplikasi akan membentuk suatu kode *survey* yang terdiri dari nama *surveyor*, nama pemilik bangunan dan alamatnya.

Aplikasi ini juga akan langsung melakukan perhitungan berdasarkan data kerusakan yang telah dimasukkan tersebut, sehingga aplikasi dapat langsung mengeluarkan hasil berupa jenis kebutuhan material beserta jumlahnya. Sehingga nantinya pihak badan penanggulangan bencana bisa dengan cepat mengetahui seberapa banyak material yang dibutuhkan, lalu dapat dengan cepat menyalurkan material tersebut kepada yang membutuhkan berdasarkan data yang

telah dimasukkan oleh surveyor tersebut. Gambaran umum mengenai aplikasi ini terlihat pada Gambar 4.2.



Gambar 4.2. Gambaran Umum Aplikasi

4.1.2. Identifikasi Aktor

Pada tahap ini bertujuan untuk melakukan identifikasi terhadap pengguna yang berhubungan dengan sistem ini dan aplikasi yang akan dibuat. Dalam sistem ini terdapat dua aktor, seperti yang dijelaskan dalam Tabel 4.1.

Tabel 4.1. Identifikasi Aktor

Aktor	Deskripsi
Pengguna (Surveyor)	Merupakan pengguna aplikasi yang dapat melakukan <i>login</i> serta mengisikan data identifikasi kerusakan ke dalam sistem.

4.1.3. Analisis Kebutuhan Fungsionalitas Sistem

Pada daftar kebutuhan fungsional terdiri dari spesifikasi kebutuhan perangkat lunak bagi pengguna, aktor yang menggunakan sistem, serta nama *use case*. Hal tersebut terlihat pada Tabel 4.2.

Tabel 4.2. Daftar Kebutuhan Fungsional

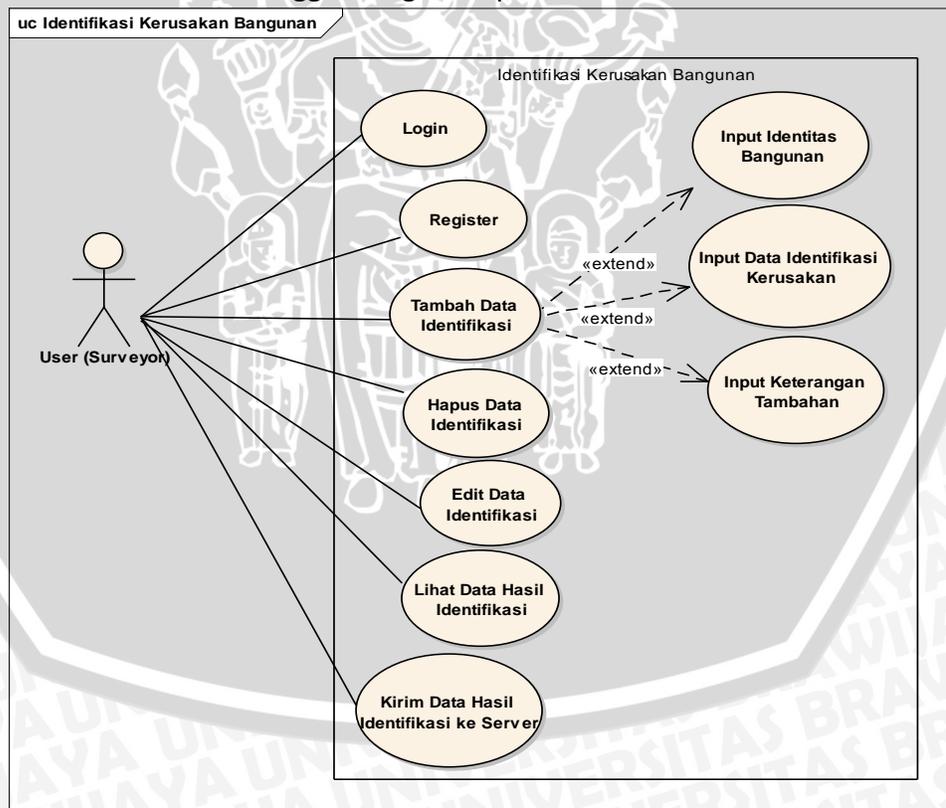
Id	Kebutuhan Fungsional	Nama Use Case
UC-01	Aplikasi mampu menampilkan halaman untuk melakukan login yang digunakan pengguna untuk masuk ke dalam sistem.	Login
UC-02	Aplikasi mampu menampilkan halaman untuk melakukan <i>register</i> yang digunakan pengguna untuk melakukan registrasi atau pendaftaran ke dalam sistem.	Register
UC-03	Aplikasi mampu menampilkan halaman untuk input data identitas bangunan yang akan dilakukan identifikasi, yaitu meliputi jenis bangunan, identitas pemilik bangunan, serta lokasi bangunan yang meliputi nama desa, kecamatan dan kabupaten atau kota, untuk input data identifikasi kerusakan bangunan yang meliputi kondisi kerusakan penutup atap, panjang dan lebar bangunan, estimasi volume kerusakan, jenis penutup atap dan foto bangunan dan untuk input keterangan tambahan yang meliputi kondisi air bersih dan keterangan lainnya	Tambah Data Identifikasi
UC-04	Aplikasi harus menyediakan fitur untuk menghapus data identifikasi kerusakan yang telah dimasukkan ke dalam basis data sistem	Hapus Data Identifikasi
UC-05	Aplikasi harus menyediakan fitur untuk melakukan edit data identifikasi kerusakan yang telah	Edit Data Identifikasi

	dimasukkan ke dalam basis data sistem	
UC-06	Aplikasi mampu menampilkan halaman untuk melihat data hasil identifikasi kerusakan bangunan	Lihat Data Hasil Identifikasi
UC-07	Aplikasi harus menyediakan fitur untuk mengirim data hasil identifikasi ke server komputer	Kirim Data Hasil Identifikasi ke Server

Selanjutnya daftar kebutuhan fungsionalitas akan dijabarkan lebih lanjut dengan *use case diagram* dan skenario *use case*.

4.1.3.1. Use Case Diagram

Use case diagram merupakan diagram yang akan menggambarkan kebutuhan dan interaksi antara user atau aktor dengan sistem pada aplikasi. Gambar 4.3 merupakan *use case diagram* dari sistem pada aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana.



Gambar 4.3. Use Case Diagram

4.1.3.2. Skenario Use Case

Dalam skenario *use case* berisi penjelasan lebih detail dari tiap-tiap *use case* yang ada pada Gambar 4.3. Pada skenario *use case*, diberikan uraian nama *use case*, aktor yang terlibat, tujuan, deskripsi, langkah-langkah *use case*, kondisi awal yang harus dipenuhi (*pre-condition*), dan kondisi akhir yang diharapkan (*post-condition*). Skenario *use case* dapat dilihat pada Tabel 4.3 sampai dengan Tabel 4.9.

Tabel 4.3. Skenario *Use Case Login*

Nama Use Case	Login
Use Case Id	UC-01
Aktor	Pengguna (Surveyor)
Tujuan	Masuk ke dalam sistem pada aplikasi
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana Pengguna mengakses menu untuk bisa masuk ke dalam sistem pada aplikasi
Pre-condition	User harus membuka aplikasi terlebih dahulu sebelum <i>Use case</i> dimulai. User memilih menu Login dari daftar menu
Post-Condition	Akan muncul halaman <i>login</i> yang berisi form isian untuk masuk ke dalam sistem aplikasi.
Aliran Utama	
Aksi dari Aktor	Respon Dari Sistem
1. User masuk ke halaman utama aplikasi dan memilih menu <i>login</i>	2. Menampilkan halaman <i>login</i> yang berisi form yang harus diisi agar bisa masuk ke dalam sistem pada aplikasi
Aliran Alternatif	
Aksi dari Aktor	Respon dari Sistem
-	-

Tabel 4.4. Skenario *Use Case Register*

Nama Use Case	Register
Use Case Id	UC-02
Aktor	Pengguna (Surveyor)
Tujuan	Melakukan registrasi atau pendaftaran ke dalam sistem aplikasi
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana Pengguna mengakses menu untuk melakukan registrasi atau pendaftaran ke dalam sistem pada aplikasi
Pre-condition	User harus membuka aplikasi terlebih dahulu sebelum <i>Use case</i> dimulai. Pengguna memilih menu login, akan tampil halaman <i>login</i> , lalu pengguna memilih menu <i>register</i> pada halaman login tersebut.
Post-Condition	Akan muncul halaman <i>register</i> yang berisi form isian untuk melakukan registrasi atau pendaftaran pada sistem aplikasi.
Aliran Utama	
Aksi dari Aktor	Respon Dari Sistem
<ol style="list-style-type: none"> 1. User masuk ke halaman utama aplikasi dan memilih menu <i>login</i> 2. Pengguna memilih menu <i>register</i> 	<ol style="list-style-type: none"> 3. Menampilkan halaman untuk melakukan registrasi atau pendaftaran yang berisi form yang harus dilengkapi oleh pengguna.
Aliran Alternatif	
Aksi dari Aktor	Respon dari Sistem
-	-

Tabel 4.5. Skenario *Use Case* Tambah Data Identifikasi

Nama Use Case	Tambah Data Identifikasi
Use Case Id	UC-03

Aktor	Pengguna (Surveyor)
Tujuan	Menambahkan data identifikasi meliputi data identitas bangunan, data identifikasi kerusakan serta data keterangan tambahan ke dalam basis data pada sistem aplikasi
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana Pengguna mengakses menu untuk menambahkan data identifikasi berupa data identitas bangunan, data identifikasi kerusakan, serta data keterangan tambahan ke dalam basis data
Pre-condition	Pengguna memilih menu tambah data identifikasi pada halaman identifikasi bangunan
Post-Condition	Form isian input input identifikasi kerusakan bangunan akan tampil.
Aliran Utama	
Aksi dari Aktor	Respon Dari Sistem
1. (a). Pengguna memilih menu tambah data identifikasi pada halaman identifikasi bangunan	2. Menampilkan form isian input identifikasi bangunan yang meliputi form isian untuk input identitas bangunan, form isian untuk input identifikasi kerusakan, serta form isian untuk input keterangan tambahan
Aliran Alternatif	
Aksi dari Aktor	Respon dari Sistem
1. (b) Pengguna memilih menu lihat data pada halaman identifikasi bangunan	-

Tabel 4.6. Skenario *Use Case* Hapus Data Identifikasi Kerusakan

Nama Use Case	Hapus Data Identifikasi Kerusakan
Use Case Id	UC-04
Aktor	Pengguna (Surveyor)
Tujuan	Menghapus data identifikasi bangunan yang sudah masuk ke dalam basis data aplikasi
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana Pengguna

	mengakses menu untuk menghapus data identifikasi kerusakan yang telah masuk ke dalam basis data
Pre-condition	Pengguna memilih data yang akan dihapus, lalu menekan data tersebut hingga muncul menu untuk menghapus data.
Post-Condition	Data identifikasi kerusakan yang dipilih akan terhapus dari basis data aplikasi.
Aliran Utama	
Aksi dari Aktor	Respon Dari Sistem
<ol style="list-style-type: none"> Pengguna memilih data identifikasi kerusakan yang akan dihapus dari basis data aplikasi Pengguna menekan data tersebut hingga muncul menu untuk hapus data Pengguna memilih menu untuk hapus data 	<ol style="list-style-type: none"> Data yang telah dipilih oleh pengguna akan dihapus dari basis data pada sistem aplikasi.
Aliran Alternatif	
Aksi dari Aktor	Respon dari Sistem
-	-

Tabel 4.7. Skenario *Use Case Edit Data Identifikasi Kerusakan*

Nama Use Case	<i>Edit Data Identifikasi Kerusakan</i>
Use Case Id	UC-05
Aktor	Pengguna (Surveyor)
Tujuan	Melakukan perubahan data identifikasi bangunan yang sudah masuk ke dalam basis data aplikasi
Deskripsi	<i>Use case ini menjelaskan bagaimana Pengguna</i>

	mengakses menu untuk merubah data identifikasi kerusakan yang telah masuk ke dalam basis data
Pre-condition	Pengguna memilih data yang akan diubah, lalu menekan data tersebut hingga muncul menu untuk merubah data.
Post-Condition	Data identifikasi kerusakan yang dipilih akan berubah sesuai dengan data yang telah diubah oleh pengguna.
Aliran Utama	
Aksi dari Aktor	Respon Dari Sistem
<ol style="list-style-type: none"> 1. Pengguna memilih data identifikasi kerusakan yang akan diubah. 2. Pengguna menekan data tersebut hingga muncul menu untuk <i>edit data</i> 3. Pengguna memilih menu untuk <i>edit data</i> 	<ol style="list-style-type: none"> 4. Data yang telah dipilih oleh pengguna akan berubah sesuai dengan data yang telah diubah oleh pengguna dan disimpan kembali ke dalam basis data sistem aplikasi.
Aliran Alternatif	
Aksi dari Aktor	Respon dari Sistem
-	-

Tabel 4.8. Skenario *Use Case* Lihat Data Hasil Identifikasi

Nama <i>Use Case</i>	Lihat Data Hasil Identifikasi
Use Case Id	UC-06
Aktor	Pengguna (Surveyor)
Tujuan	Melihat data kebutuhan material berdasarkan isian data identifikasi kerusakan bangunan
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana Pengguna mengakses menu untuk

	melihat hasil dari pengolahan isian data identifikasi kerusakan yaitu berupa data kebutuhan material
Pre-condition	Pengguna memilih menu untuk lihat data pada halaman identifikasi bangunan
Post-Condition	Halaman lihat data hasil identifikasi kerusakan bangunan akan tampil
Aliran Utama	
Aksi dari Aktor	Respon Dari Sistem
1. Pengguna memilih menu lihat data hasil identifikasi kerusakan pada halaman identifikasi bangunan	2. Menampilkan data hasil identifikasi kerusakan bangunan berdasarkan pengolahan dari hasil isian data identifikasi yang telah diisikan sebelumnya
Aliran Alternatif	
Aksi dari Aktor	Respon dari Sistem
-	-

Tabel 4.9. Skenario *Use Case* Kirim Data Hasil Identifikasi ke Server

Nama Use Case	Kirim Data Hasil Identifikasi ke Server
Use Case Id	UC-07
Aktor	Pengguna (Surveyor)
Tujuan	Mengirimkan data hasil identifikasi menuju ke basis data pada server komputer.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana Pengguna mengakses menu untuk mengirim data hasil identifikasi menuju ke dalam basis data pada server komputer
Pre-condition	User memilih menu <i>sync</i> pada halaman lihat data
Post-Condition	Data yang terdapat pada basis data aplikasi perangkat bergerak akan dikirimkan menuju ke server pada komputer.
Aliran Utama	
Aksi dari Aktor	Respon Dari Sistem
1. Pengguna memilih menu tambah data identifikasi pada halaman	4. Data yang terdapat di dalam basis data perangkat bergerak akan dikirimkan menuju ke basis data pada server komputer.

identifikasi bangunan	
2. Pengguna menggeser halaman untuk menampilkan halaman lihat data	
3. Pengguna memilih tombol sync	
Aliran Alternatif	
Aksi dari Aktor	Respon dari Sistem
-	-

4.1.4. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan suatu kebutuhan yang mana kebutuhan tersebut tidak diminta oleh pengguna, namun dibutuhkan oleh pengguna tersebut. Kebutuhan non fungsional pada aplikasi ini dijelaskan pada Tabel 4.10.

Tabel 4.10. Spesifikasi Kebutuhan Non Fungsional

Parameter	Deskripsi Kebutuhan
<i>Compatibility</i>	Aplikasi minimal dapat di jalan kan pada sistem operasi Android 4.4 (<i>KitKat</i>)
<i>Usability</i>	Rancangan antarmuka mudah digunakan oleh pengguna biasa dengan kaidah <i>usability</i> seperti <i>learnability</i> , <i>efficiency</i> , <i>memorability</i> , <i>error</i> dan <i>satisfication</i> . Dengan target tingkat usabilitas 80 %

4.2. Perancangan Aplikasi Perangkat Bergerak

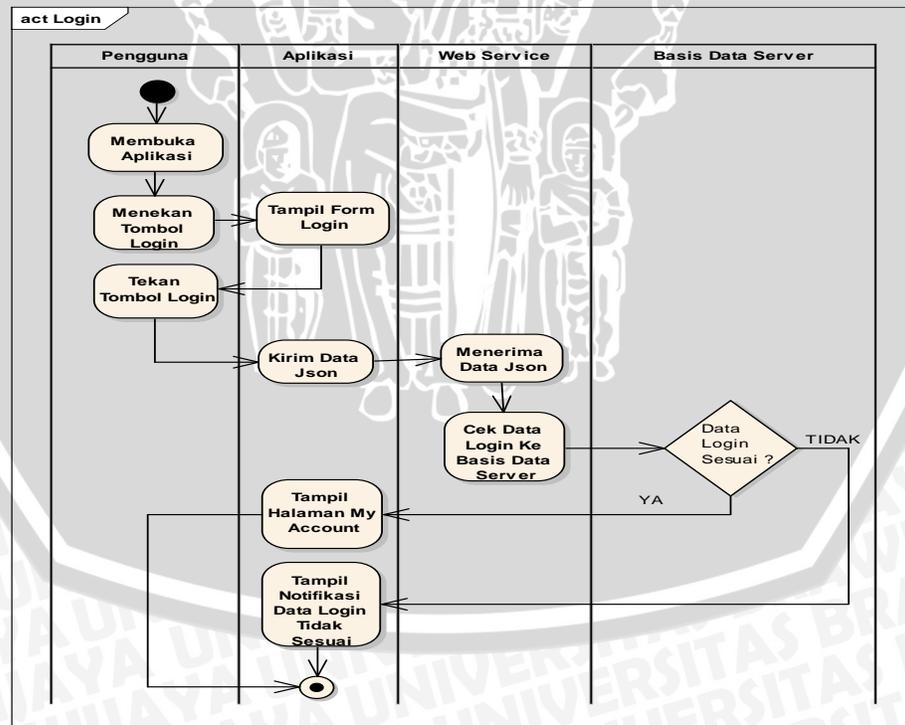
Perancangan aplikasi perangkat bergerak sendiri terdiri dari beberapa tahapan yaitu perancangan arsitektur sistem, perancangan aktivitas, perancangan kelas, perancangan basis data, dan perancangan navigasi dan antarmuka. Tahap perancangan arsitektur sistem menjelaskan tentang arsitektur sistem yang akan dibuat dan dikembangkan. Tahap perancangan basis data menjelaskan entitas dan relasinya masing-masing dan tahap

perancangan navigasi dan antarmuka mendefinisikan tampilan dan pengalaman antar muka yang akan diimplementasi.

4.2.1. Perancangan Activity Diagram

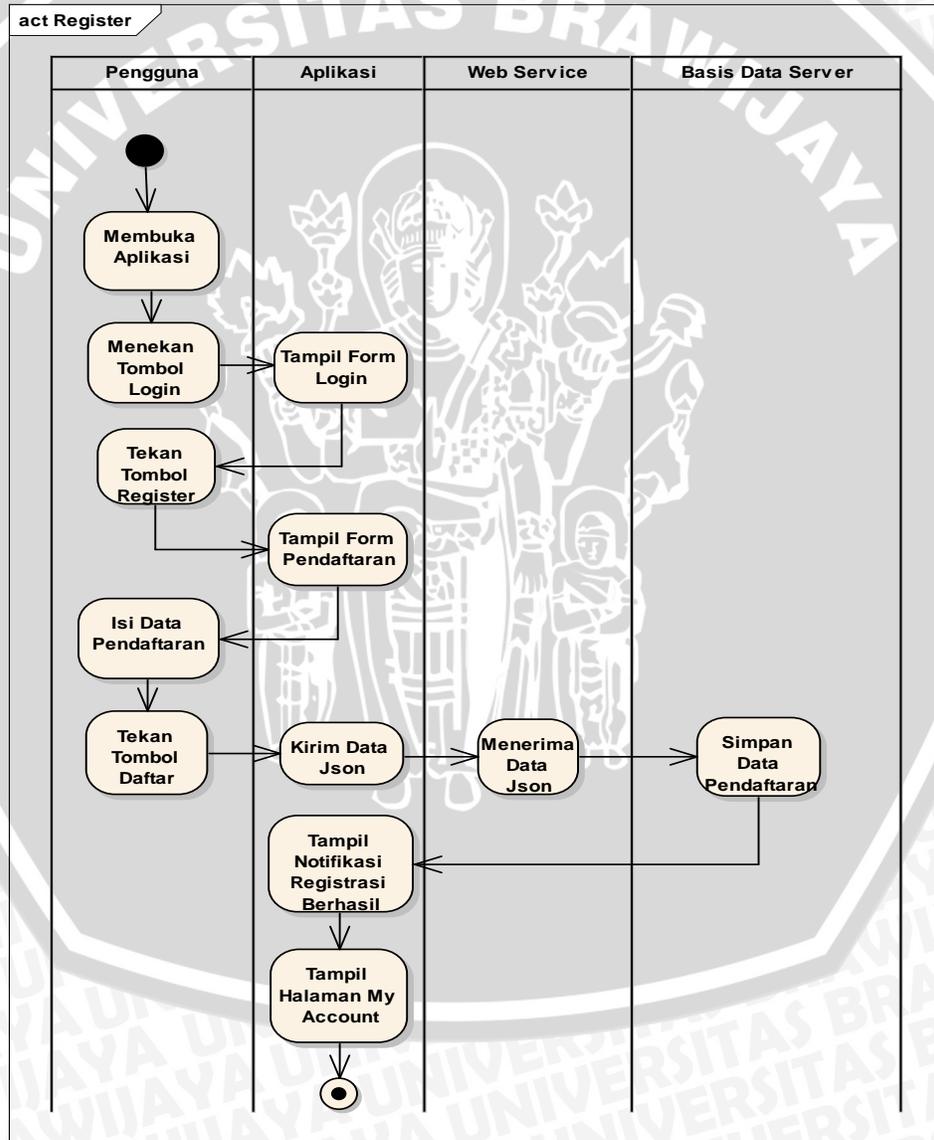
Perancangan *activity diagram* merupakan pemodelan aktivitas pengguna dengan sistem yang berjalan berdasarkan pada skenario *use case*. Pada *activity diagram* alur kerja berupa langkah dan aksi dari kebutuhan fungsional dijabarkan dalam bentuk grafis. Di dalam *activity diagram* juga terdapat gambaran perulangan dan pilihan yang akan dijalankan. *Activity diagram* pada aplikasi ini dijelaskan pada Gambar 4.4 sampai Gambar 4.10.

Pada Gambar 4.4 menampilkan aktifitas dari pengguna, jika pengguna tersebut ingin melakukan *login* untuk masuk ke dalam aplikasi. Pertama, pengguna harus menekan tombol *login* yang terdapat pada halaman utama aplikasi. Lalu aplikasi akan merespon dengan menampilkan form *login*. Setelah itu pengguna akan mengisi form tersebut, jika sudah selesai mengisi form tersebut, pengguna akan menekan tombol *login*. Lalu aplikasi akan mengecek apakah data yang dimasukkan oleh pengguna tersebut telah sesuai dengan data yang ada di dalam basis data atau tidak. Jika telah sesuai, maka pengguna akan diarahkan menuju ke halaman *myaccount*.



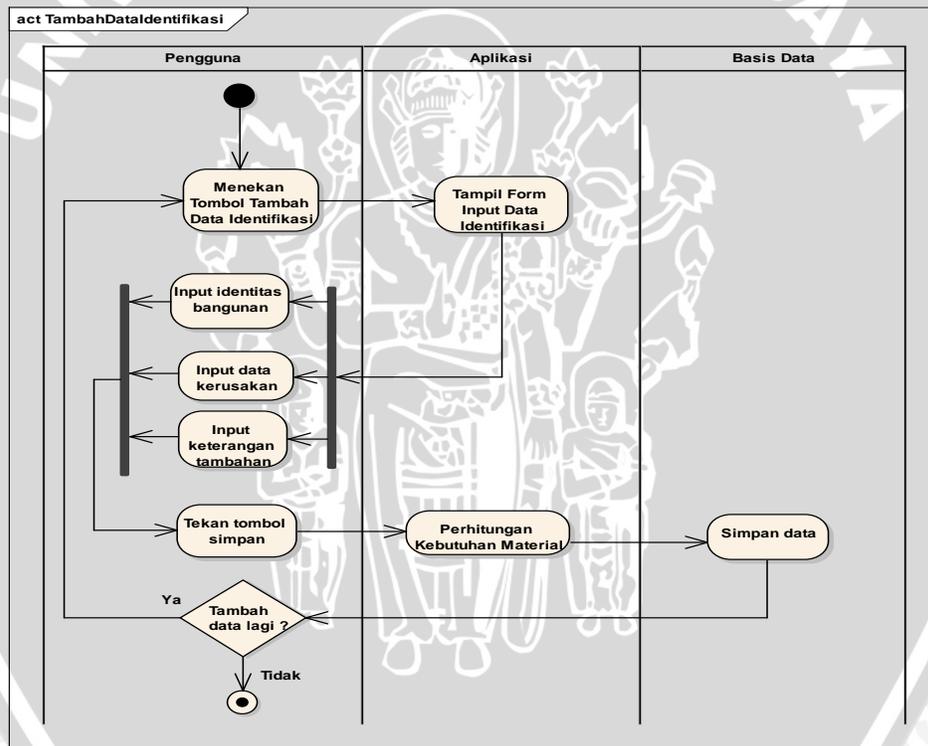
Gambar 4.4. Activity Diagram Login

Pada Gambar 4.5 menampilkan aktifitas dari pengguna, jika pengguna tersebut ingin melakukan *register* ke dalam aplikasi. Pertama, pengguna harus menekan tombol *login* yang terdapat pada halaman utama aplikasi. Lalu aplikasi akan merespon dengan menampilkan form *login*. Setelah itu pengguna akan menekan tombol *register* yang terdapat pada form tersebut. Lalu pengguna akan diarahkan menuju halaman pendaftaran yang berisi form pendaftaran. Pengguna akan mengisi form pendaftaran tersebut, jika sudah selesai mengisi form tersebut, pengguna akan menekan tombol *register*. Lalu aplikasi akan mengecek apakah data yang dimasukkan oleh pengguna tersebut telah sesuai. Jika telah sesuai, maka pengguna akan diarahkan menuju ke halaman *login*.



Gambar 4.5. Activity Diagram Register

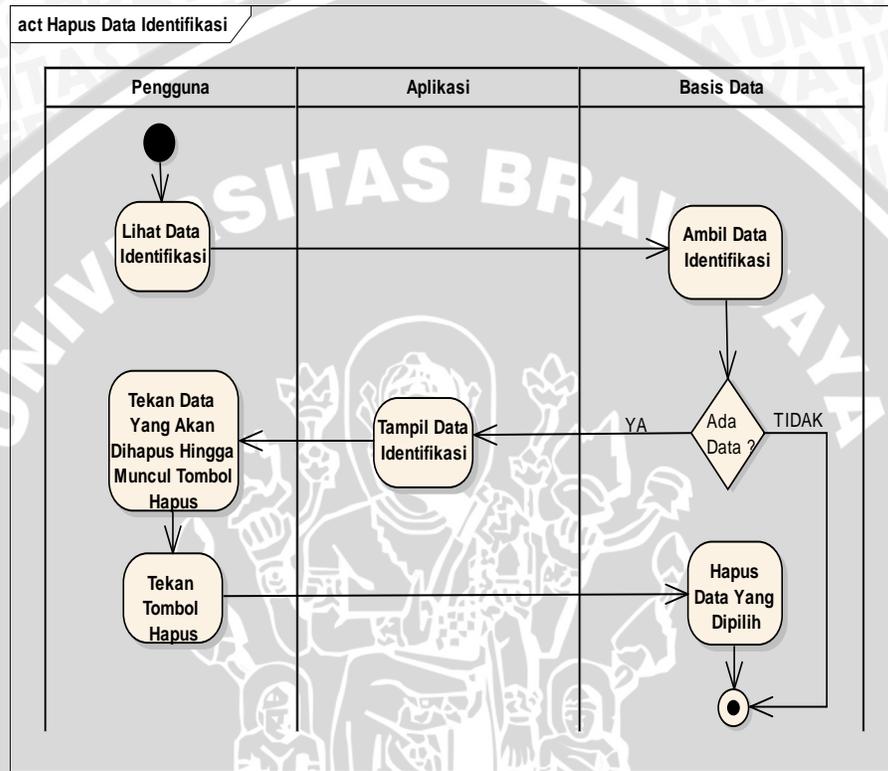
Pada Gambar 4.6 menampilkan aktifitas dari pengguna, jika pengguna tersebut ingin melakukan penambahan data identifikasi pada aplikasi. Pertama, pengguna harus menekan tombol tambah data identifikasi yang terdapat pada aplikasi. Lalu aplikasi akan merespon dengan menampilkan halaman untuk melakukan *input* data identifikasi, yang mana pada halaman tersebut terdapat beberapa form seperti form *input* identitas bangunan, form *input* data kerusakan, dan form *input* keterangan tambahan. Setelah itu pengguna akan mengisi form data identifikasi yang terdapat pada halaman tersebut. Setelah selesai melakukan pengisian data identifikasi, pengguna harus menekan tombol simpan agar data yang dimasukkan oleh pengguna bisa tersimpan di dalam database pada sistem dalam aplikasi. Setelah melakukan penyimpanan data, pengguna bisa menambahkan data baru lagi dengan menekan tombol tambah data identifikasi seperti yang telah dilakukan sebelumnya.



Gambar 4.6. Activity Diagram Tambah Data Identifikasi

Pada Gambar 4.7 menampilkan aktifitas dari pengguna, jika pengguna tersebut ingin menghapus data hasil identifikasi kerusakan bangunan yang terdapat dalam basis data. Untuk menghapus data hasil identifikasi kerusakan bangunan, pengguna pertama harus memilih menu lihat data yang terdapat pada pilihan menu di aplikasi. Setelah pengguna memilih menu tersebut, sistem pada aplikasi akan merespon dengan melakukan

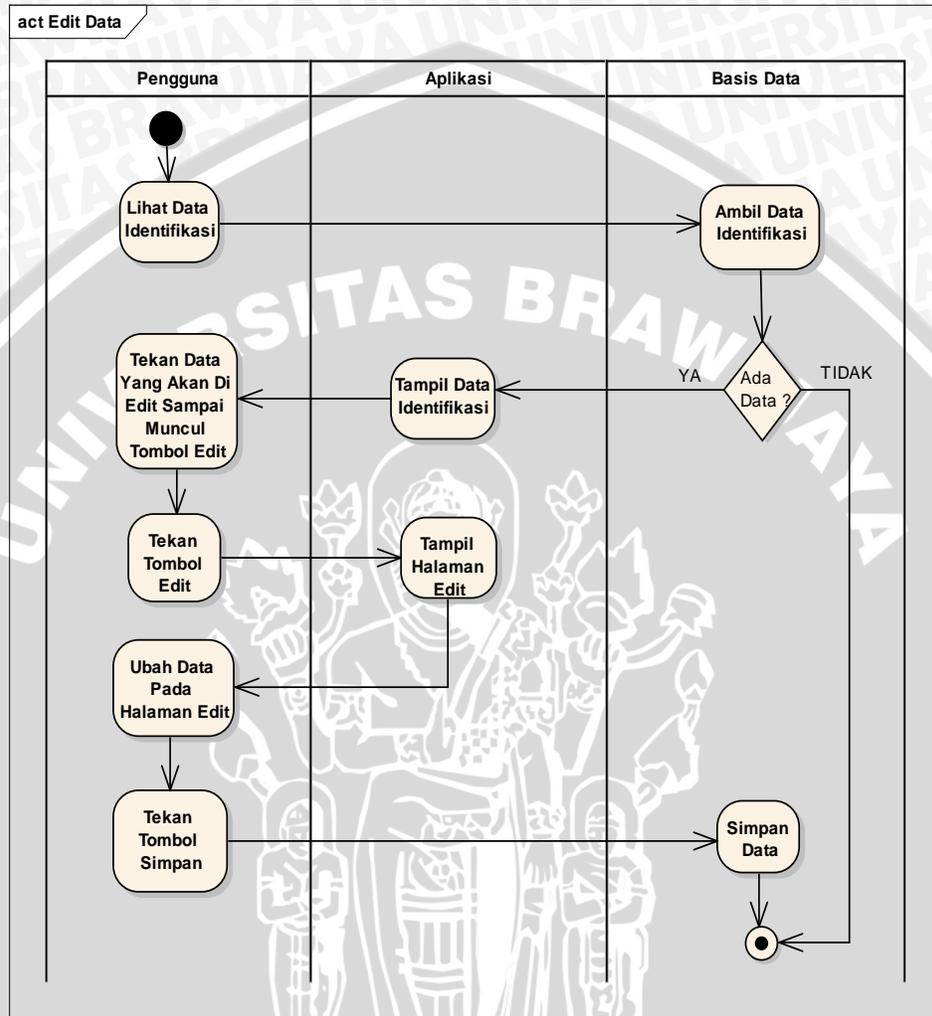
pengecekan pada basis data. Jika terdapat data kebutuhan material pada basis data, maka seluruh data tersebut akan ditampilkan pada aplikasi. Lalu pengguna akan memilih data mana yang akan dihapus, dan menekan data tersebut hingga muncul menu untuk hapus data. Setelah itu pengguna akan menekan tombol hapus data, dan data tersebut pun akan dihapus dari basis data.



Gambar 4.7. Activity Diagram Hapus Data Identifikasi

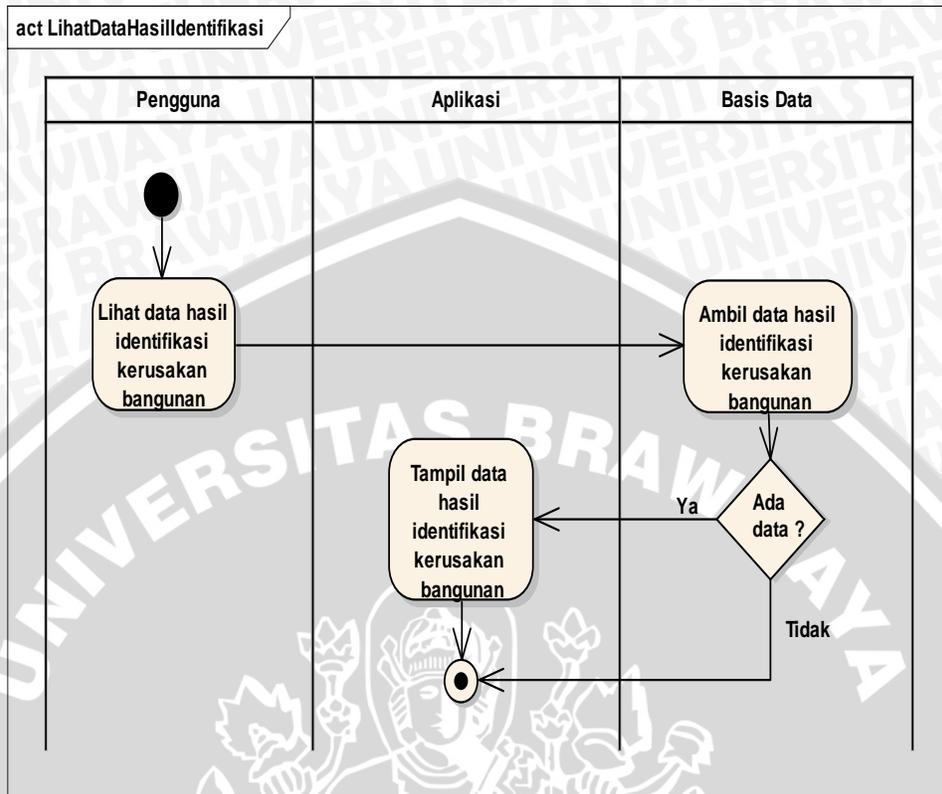
Pada Gambar 4.8 menampilkan aktifitas dari pengguna, jika pengguna tersebut ingin melakukan perubahan pada data hasil identifikasi kerusakan bangunan yang telah masuk ke dalam basis data. Untuk merubah data hasil identifikasi kerusakan bangunan, pengguna pertama harus memilih menu lihat data yang terdapat pada pilihan menu di aplikasi. Setelah pengguna memilih menu tersebut, sistem pada aplikasi akan merespon dengan melakukan pengecekan pada basis data. Jika terdapat data kebutuhan material pada basis data, maka seluruh data tersebut akan ditampilkan pada aplikasi. Lalu pengguna akan memilih data mana yang akan dirubah, dan menekan data tersebut hingga muncul menu untuk *edit* data. Setelah itu pengguna akan menekan tombol *edit* data. Lalu pengguna akan diarahkan menuju halaman edit data, yang berisi form data identifikasi yang telah diisi sebelumnya oleh pengguna

tersebut. Setelah selesai merubah data, pengguna harus menekan tombol simpan, untuk menyimpan data yang telah dirubah tersebut ke dalam basis data.



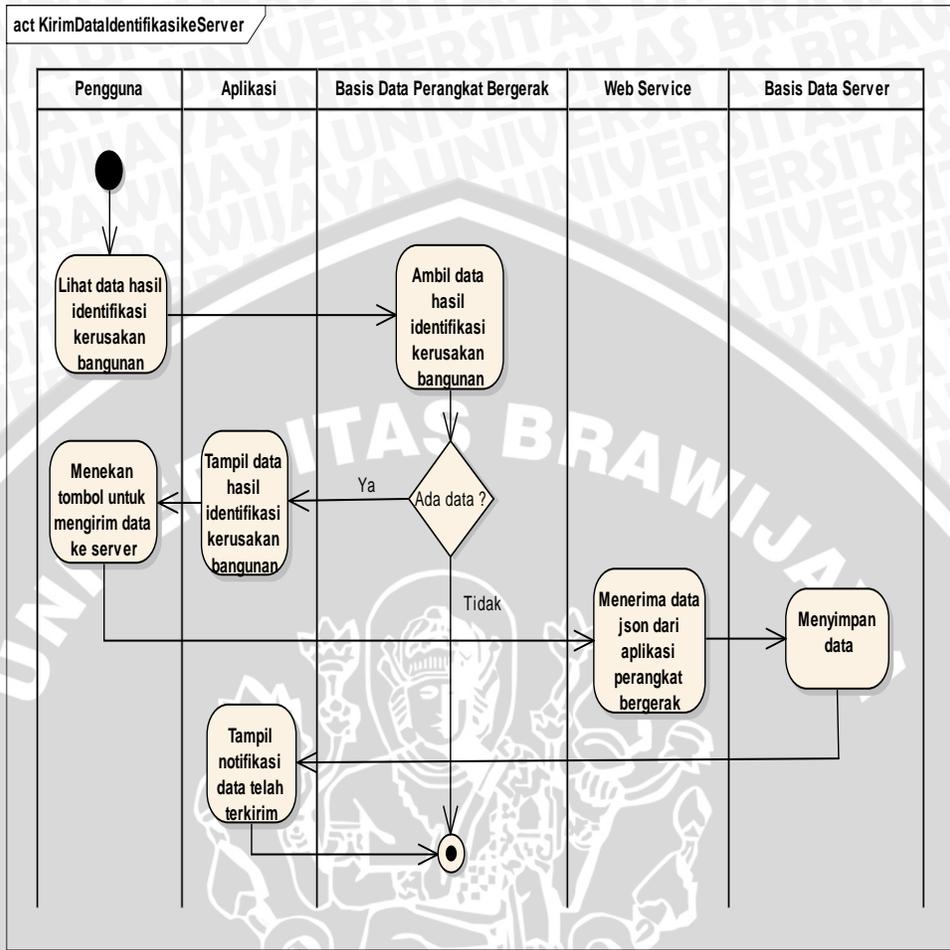
Gambar 4.8. Activity Diagram Edit Data Identifikasi

Pada Gambar 4.9 menampilkan aktifitas dari pengguna, jika pengguna tersebut ingin melihat ataupun menampilkan data hasil identifikasi kerusakan bangunan. Untuk melihat data hasil identifikasi kerusakan bangunan, pengguna pertama harus memilih menu lihat data yang terdapat pada pilihan menu di aplikasi. Setelah pengguna memilih menu tersebut, sistem pada aplikasi akan merespon dengan melakukan pengecekan pada basis data. Jika terdapat data kebutuhan material pada basis data, maka seluruh data tersebut akan ditampilkan pada aplikasi.



Gambar 4.9. Activity Diagram Lihat Data Hasil Identifikasi Kerusakan Bangunan

Pada Gambar 4.10 menampilkan aktifitas dari pengguna, jika pengguna tersebut ingin mengirimkan data hasil identifikasi kerusakan bangunan menuju ke server pada komputer. Untuk mengirimkan data hasil identifikasi kerusakan bangunan, pengguna pertama harus memilih menu lihat data yang terdapat pada pilihan menu di aplikasi. Setelah pengguna memilih menu tersebut, sistem pada aplikasi akan merespon dengan melakukan pengecekan pada basis data. Jika terdapat data identifikasi bangunan pada basis data, maka seluruh data tersebut akan ditampilkan pada aplikasi. Setelah itu pengguna harus menekan tombol untuk mengirimkan data hasil identifikasi yang terdapat pada halaman tersebut. Data tersebut akan dirubah terlebih dahulu oleh aplikasi dalam bentuk json, setelah itu baru di kirimkan menuju ke *web service*, lalu oleh *web service* data tersebut akan dimasukkan ke dalam basis data yang terdapat di dalam *server* komputer. Setelah itu *web service* akan mengirimkan data *json* ke aplikasi sebagai pemberitahuan bahwa data telah masuk ke basis data yang terdapat di dalam *server* komputer.



Gambar 4.10. Activity Diagram Kirim Data Hasil Identifikasi Kerusakan Bangunan ke Server

4.2.2. Perancangan Arsitektur Sistem

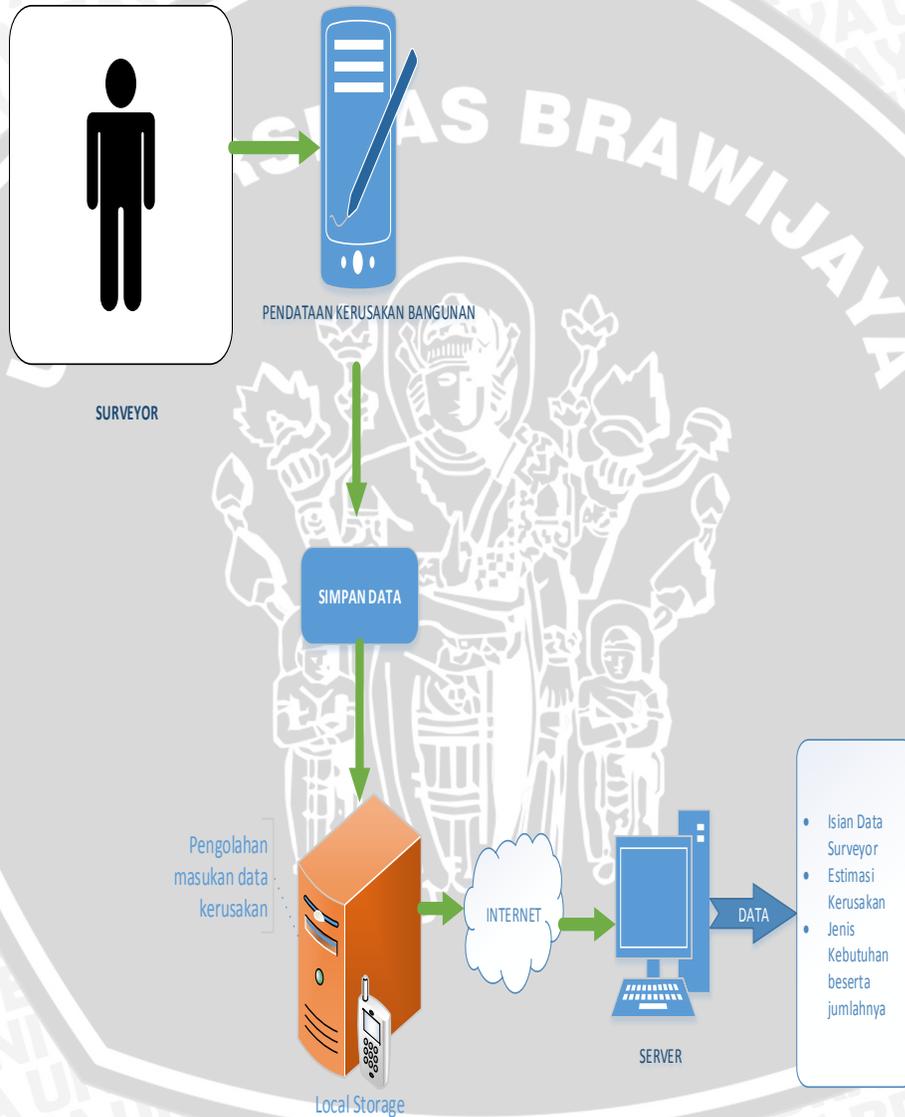
Perancangan arsitektur pada sistem aplikasi Identifikasi dan Klasifikasi Kerusakan Rumah Warga Dampak Bencana dapat digambarkan seperti pada Gambar 4.11.

Penjelasan pada Gambar 4.11 yaitu, pertama surveyor akan menuju ke lokasi bencana untuk melakukan pendataan kerusakan bangunan, surveyor melakukan pendataan kerusakan dengan memasukkan data tersebut ke dalam form isian yang terdapat pada aplikasi, data tersebut meliputi data identitas bangunan, data identifikasi kerusakan, serta data keterangan tambahan.

Semua data yang diinputkan oleh surveyor tersebut didapatkan berdasarkan hasil pengamatan langsung secara *visual* di lapangan. Setelah selesai memasukkan data pada aplikasi, surveyor akan menekan tombol simpan, sehingga data yang telah dimasukkan akan tersimpan di dalam

media penyimpanan pada perangkat bergerak yang digunakan oleh surveyor untuk melakukan input data.

Di dalam sistem aplikasi pada perangkat bergerak tersebut data akan dilakukan pengolahan. Sehingga jika data tersebut diunggah ke server melalui internet, maka data yang akan diterima oleh server berupa data hasil pengolahan dari data masukan surveyor, yaitu seperti data jenis material beserta jumlah yang dibutuhkan.

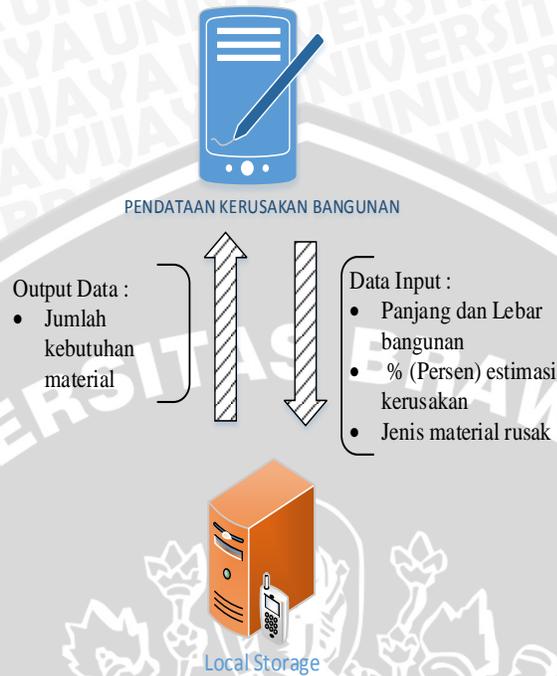


Gambar 4.11. Arsitektur Sistem

4.2.2.1. Perancangan Pengolahan Data Identifikasi

Pada perancangan pengolahan data identifikasi ini akan menjelaskan mengenai bagaimana data hasil masukan surveyor bisa menghasilkan

data berupa jenis material beserta jumlah yang dibutuhkan. Perancangan tersebut dapat terlihat seperti pada Gambar 4.12.



Gambar 4.12. Rancangan Pengolahan Data Identifikasi

Pada Gambar 4.12 dapat dijelaskan bahwa terdapat data *input* berupa Panjang dan Lebar bangunan, % (persen) estimasi kerusakan berdasarkan tampak *visual*, serta jenis bahan dari material yang rusak. Berdasarkan data *input* tersebut, nantinya dapat dihasilkan *output* berupa jumlah material yang dibutuhkan untuk mengganti kerusakan yang timbul tersebut.

Contohnya seperti pada saat terdapat kerusakan pada penutup atap bangunan rumah warga, surveyor akan melakukan pengukuran panjang dan lebar dari bangunan tersebut, lalu surveyor akan memasukkan data panjang dan lebar tersebut ke dalam aplikasi. Sistem pada aplikasi akan otomatis mengolah data masukan tersebut, sehingga akan dihasilkan volume luas dari bangunan tersebut. Selain masukan panjang dan lebar dari bangunan tersebut, surveyor juga memasukkan data berupa berapa % (persen) estimasi kerusakan yang terdapat pada penutup atap tersebut. Terakhir surveyor juga akan memasukkan data jenis material yang rusak tersebut. Seperti contohnya, kerusakan terjadi pada penutup atap berupa genteng yang berjenis genteng *press*, yang mana pada basis data sistem aplikasi sudah terdapat data asumsi jumlah material penutup atap tiap m^2 , yang mana setiap jenis penutup atap memiliki asumsi jumlah yang berbeda tiap m^2 . Contohnya pada penutup atap berjenis genteng *press*

tersebut membutuhkan 25 buah untuk menutup setiap 1 m² atap bangunan. Setelah semua data yang dibutuhkan sistem telah selesai diinputkan, sistem akan mulai melakukan proses perhitungan. Perhitungan tersebut dapat terlihat seperti pada contoh studi kasus sebagai berikut :

Input:

$$p = 10$$

$$l = 13.5$$

Jenis penutup atap = Genteng *press*

% kerusakan = 25%

Perhitungan :

$$\begin{aligned} Lb &= p * l \\ &= 10 * 13.5 \\ &= 135m^2 \end{aligned}$$

$$\begin{aligned} La &= 1.1 * Lb \\ &= 1.1 * 135 \\ &= 148.5m^2 \end{aligned}$$

$$\begin{aligned} Vr &= La * \% \text{ rusak} \\ &= 148.5 * 25\% \end{aligned}$$

$$= 37.125m^2$$

$$Jm = Vr * Am$$

$$= 37.125 * 25$$

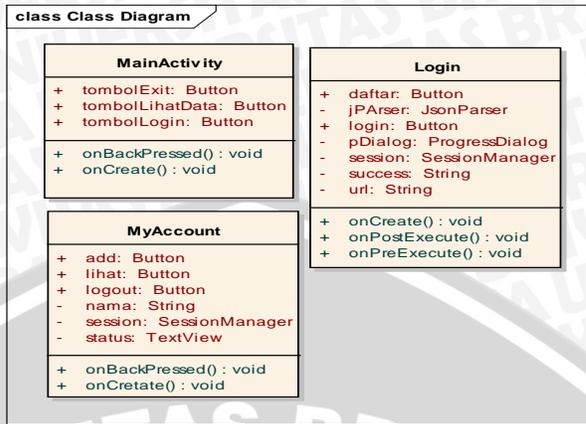
$$= 928.125 \text{ Buah}$$

$$= 929 \text{ Buah}$$

Dari contoh perhitungan tersebut, maka akan didapatkan hasil berupa jumlah kebutuhan material berupa genteng *press* sebanyak 929 buah.

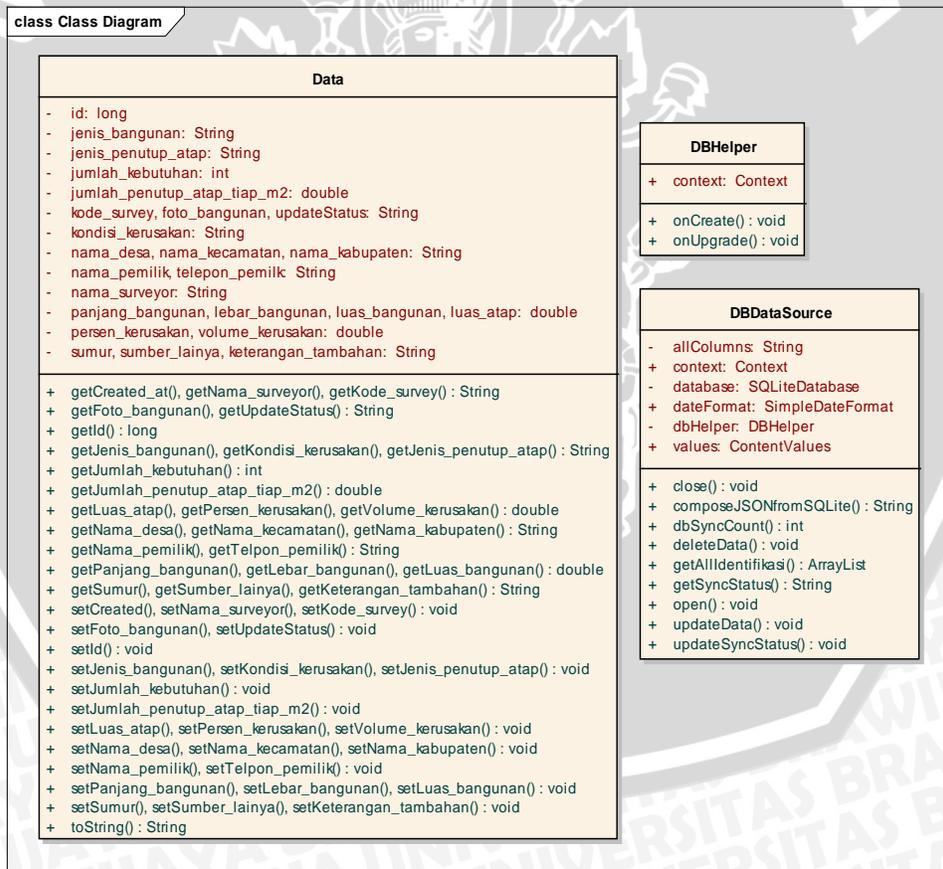
4.2.3. Perancangan *Class Diagram*

Perancangan *class diagram* digunakan untuk menampilkan beberapa kelas yang ada pada sistem yang akan dibangun. *Class diagram* merepresentasikan hubungan antar kelas di dalam sistem dan bagaimana cara antar kelas saling terhubung. Penjelasan kelas yang digunakan pada aplikasi perangkat bergerak berupa nama atribut dan operasi yang digunakan dalam setiap kelas dapat dilihat pada Gambar 4.13 sampai Gambar 4.17.



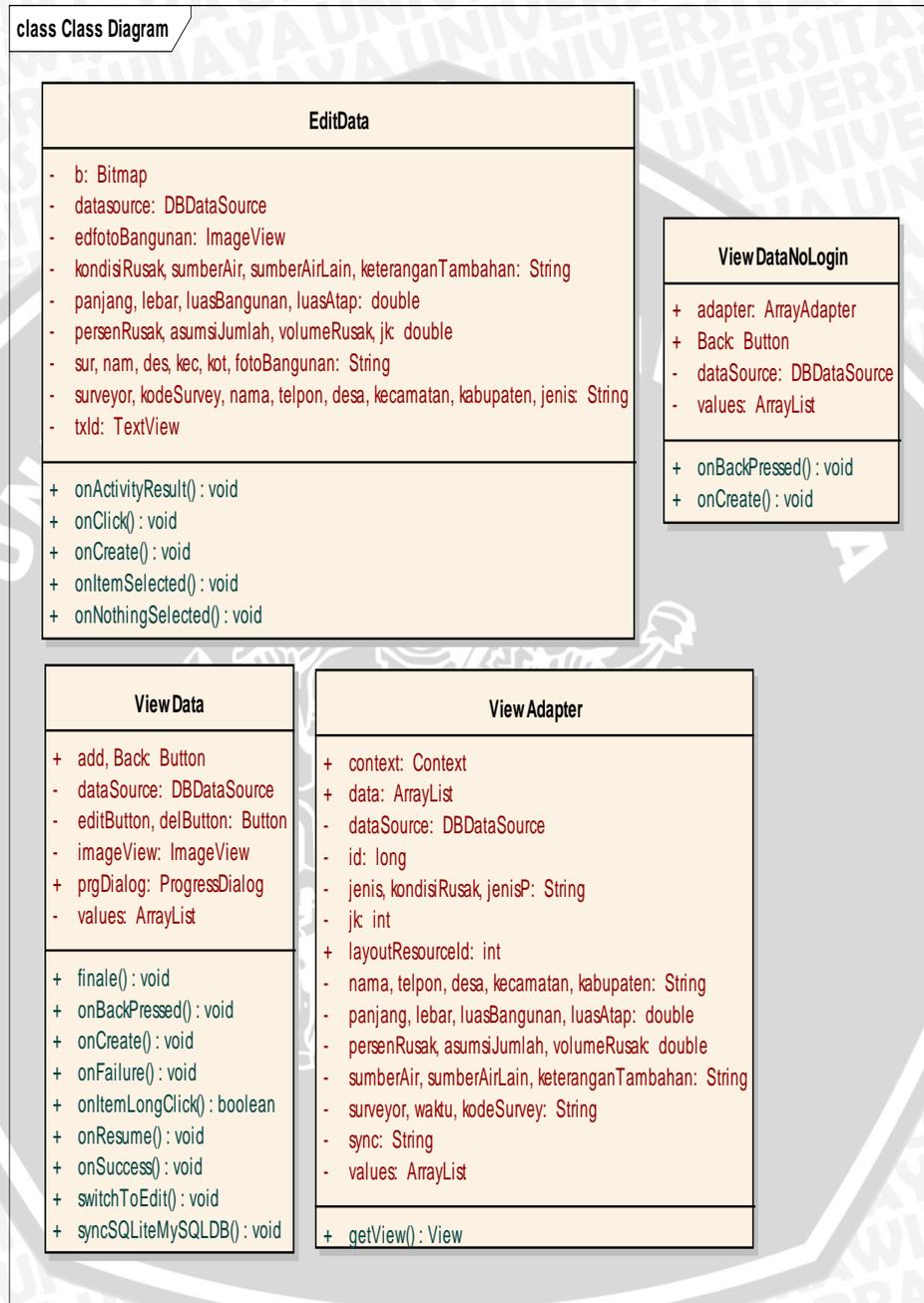
Gambar 4.13. Penjelasan Kelas pada Kelas *Diagram* Perangkat Bergerak (Bagian 1)

Pada Gambar 4.13 menjelaskan beberapa kelas *activity* yang dirancang pada aplikasi perangkat bergerak yang akan dibangun. Kelas *activity* tersebut, yaitu : *MainActivity*, *Login*, dan *MyAccount*.



Gambar 4.14. Penjelasan Kelas pada Kelas *Diagram* Perangkat Bergerak (Bagian 2)

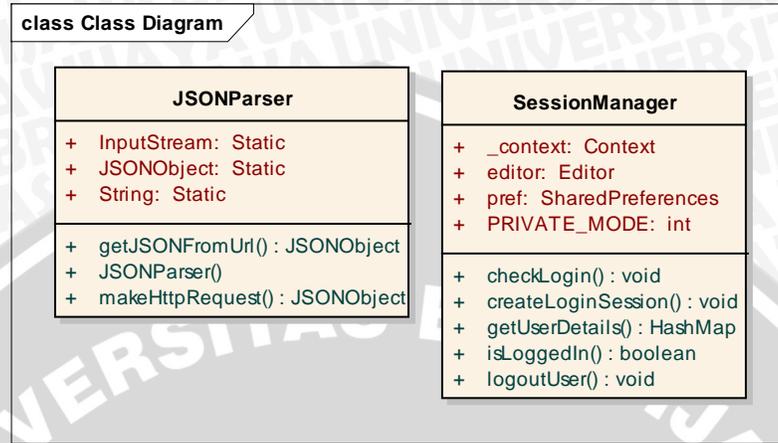
Pada Gambar 4.14 menjelaskan beberapa kelas *activity* yang dirancang pada aplikasi perangkat bergerak yang akan dibangun. Kelas *activity* tersebut, yaitu : *Data*, *DBHelper*, dan *DBDataSource*.



Gambar 4.15. Penjelasan Kelas pada Kelas *Diagram* Perangkat Bergerak (Bagian 3)

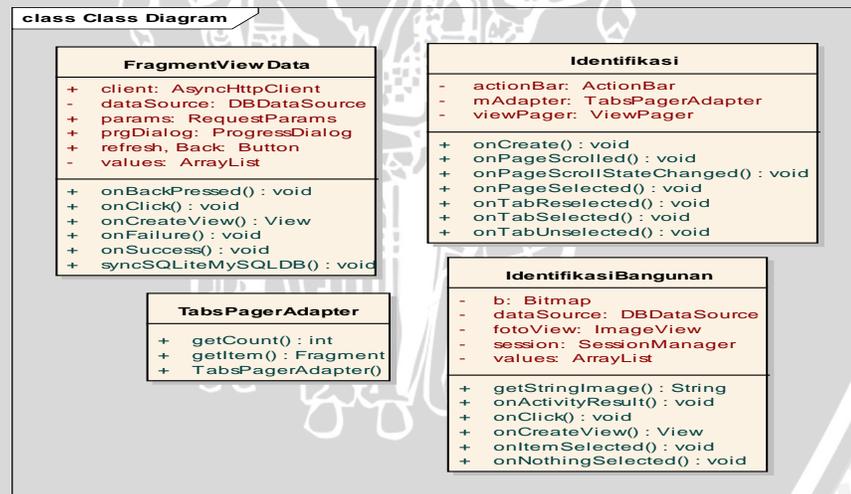
Pada Gambar 4.15 menjelaskan beberapa kelas *activity* yang dirancang pada aplikasi perangkat bergerak yang akan dibangun. Kelas

activity tersebut, yaitu : *EditData*, *ViewData*, *ViewAdapter*, dan *ViewDataNoLogin*



Gambar 4.16. Penjelasan Kelas pada Kelas *Diagram* Perangkat Bergerak (Bagian 4)

Pada Gambar 4.16 menjelaskan beberapa kelas *activity* yang dirancang pada aplikasi perangkat bergerak yang akan dibangun. Kelas *activity* tersebut, yaitu : *JsonParser*, dan *SessionManager*



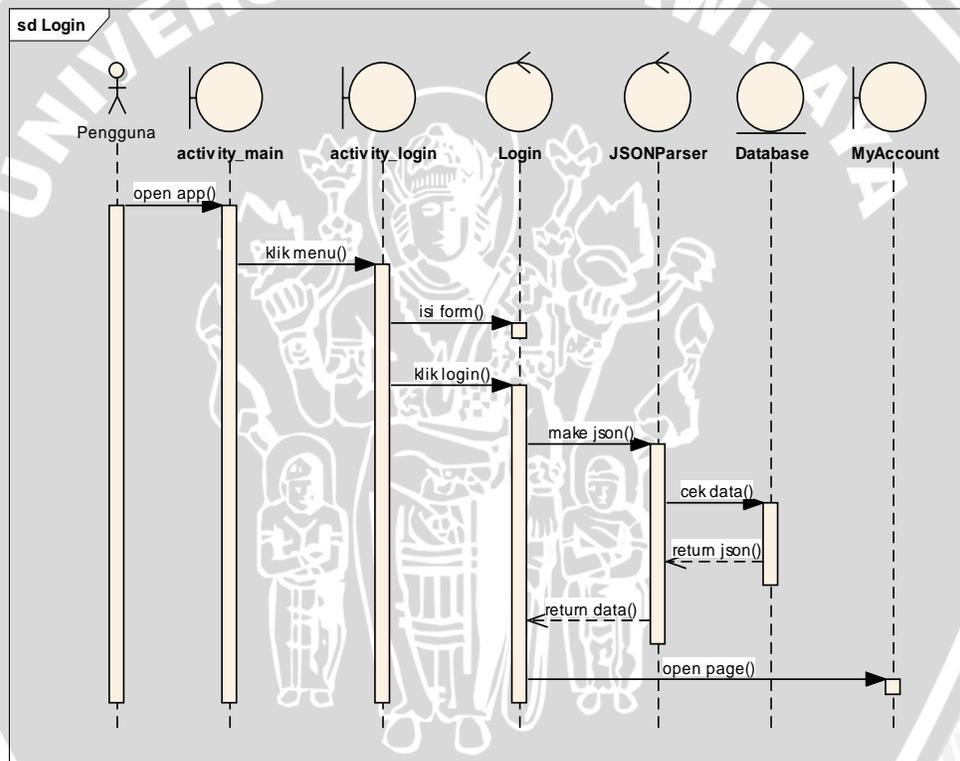
Gambar 4.17. Penjelasan Kelas pada Kelas *Diagram* Perangkat Bergerak (Bagian 5)

Pada Gambar 4.17 menjelaskan beberapa kelas *activity* yang dirancang pada aplikasi perangkat bergerak yang akan dibangun. Kelas *activity* tersebut, yaitu : *FragmentViewData*, *Identifikasi*, *IdentifikasiBangunan*, dan *TabsPagerAdapter*

4.2.4. Perancangan *Sequence Diagram*

Perancangan *sequence diagram* menjelaskan interaksi dan hubungan antar objek dan kelas yang dirancang.

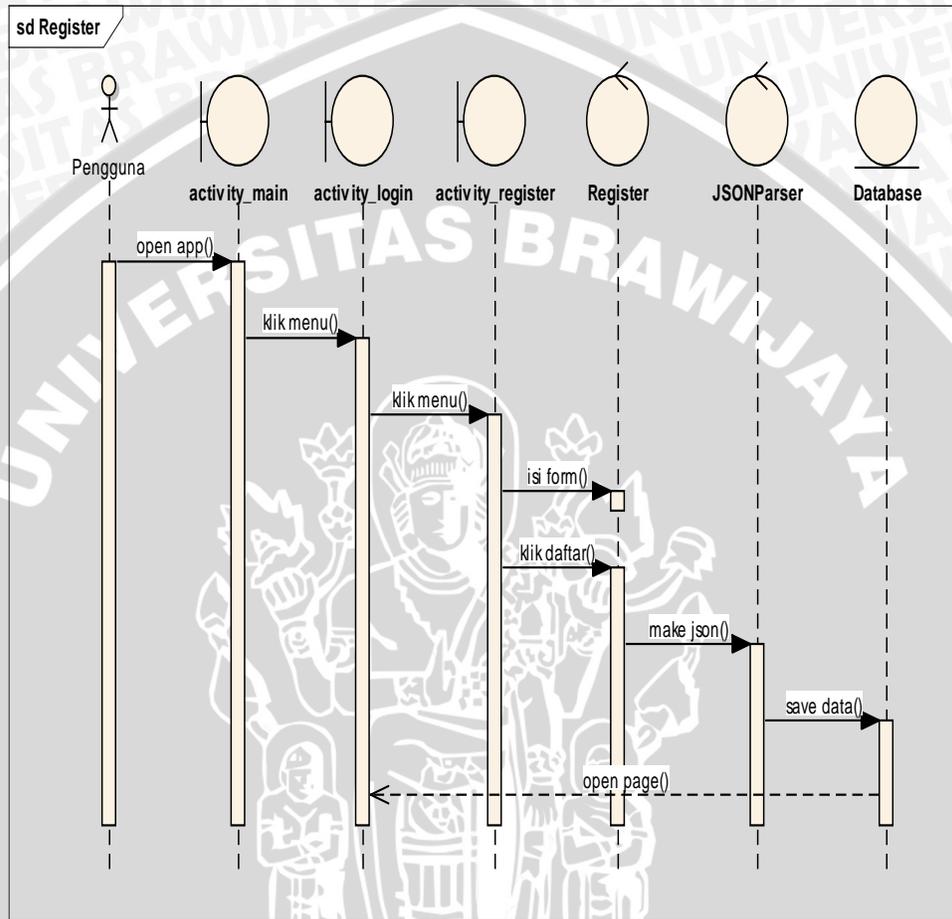
Pada Gambar 4.18 akan menampilkan *sequence diagram login*. Pertama pengguna membuka aplikasi, lalu halaman *activity_main* akan tampil, lalu pengguna menekan tombol login, setelah itu akan diarahkan menuju ke halaman login, pada halaman tersebut pengguna mengisi alamat email dan *password* pada form yang telah disediakan, setelah itu menekan tombol *login*, lalu aplikasi akan mengecek alamat email dan *password* yang telah dimasukkan oleh pengguna, di basis data server melalui kelas *JSONParser*, jika *login* sukses, maka pengguna akan diarahkan menuju ke halaman *MyAccount*.



Gambar 4.18. *Sequence Diagram Login*

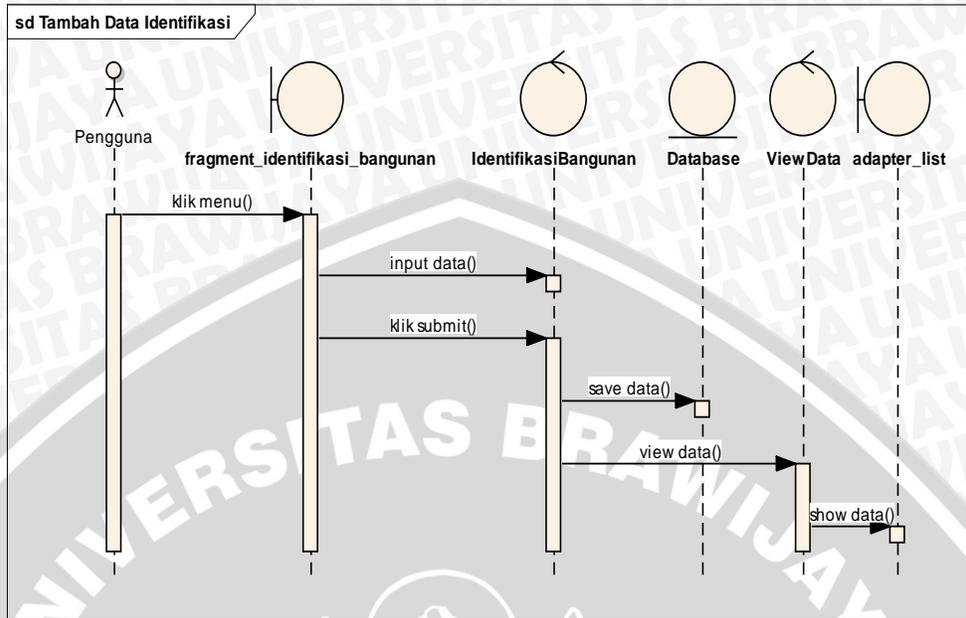
Pada Gambar 4.19 akan menampilkan *sequence diagram register*. Pertama pengguna membuka aplikasi, lalu halaman *activity_main* akan tampil, lalu pengguna menekan tombol login, setelah itu akan diarahkan menuju ke halaman login, setelah itu pengguna menekan tombol *register*. Lalu pengguna akan diarahkan menuju halaman pendaftaran. Pada halaman tersebut pengguna mengisi nama lengkap, alamat email dan *password*, setelah itu klik tombol daftar, lalu aplikasi akan mengirimkan nama lengkap, alamat email dan *password* yang telah dimasukkan oleh

pengguna, ke dalam kelas *JSONParser*, lalu kelas tersebut akan mengubah data tersebut menjadi data json lalu mengirimnya ke basis data pada server untuk disimpan, jika pendaftaran sukses, maka pengguna akan diarahkan menuju ke halaman *login*.



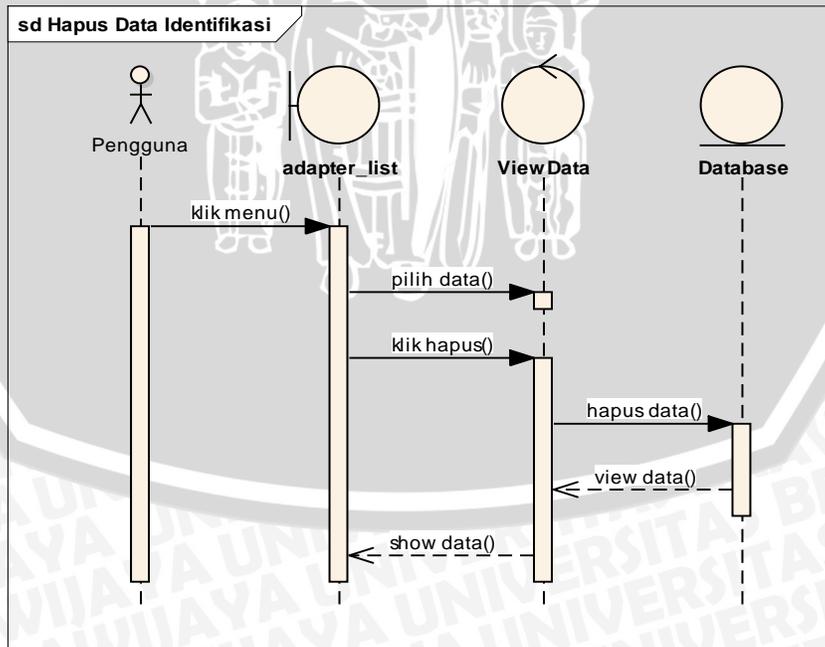
Gambar 4.19. Sequence Diagram Register

Pada Gambar 4.20 akan menampilkan *sequence diagram* tambah data identifikasi. Pertama pengguna akan menekan tombol untuk menambahkan data identifikasi. Lalu pengguna akan diarahkan menuju ke halaman *fragment_identifikasi_bangunan*. Pada halaman tersebut pengguna akan mengisikan data identifikasi kerusakan bangunan pada form yang telah disediakan. Setelah itu pengguna akan menekan tombol *submit*. Lalu data yang telah diisikan tersebut akan di kirim ke dalam basis data pada sistem aplikasi untuk disimpan, jika proses tersebut berhasil, maka pengguna akan diarahkan ke halaman lihat data, untuk melihat data yang telah dimasukkan tersebut.



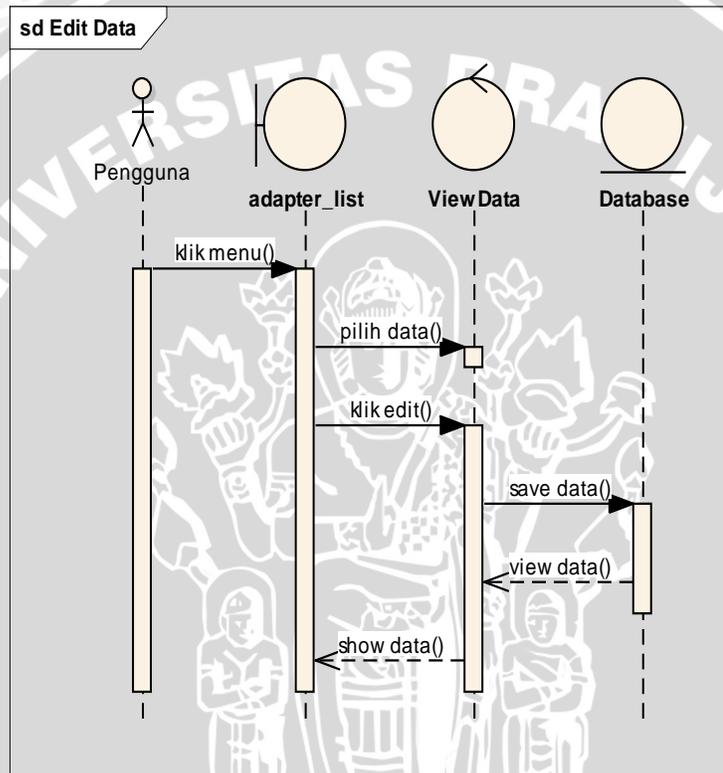
Gambar 4.20. Sequence Diagram Tambah Data Identifikasi

Pada Gambar 4.21 akan menampilkan *sequence diagram* hapus data identifikasi. Pertama pengguna masuk ke halaman lihat data, lalu pengguna memilih data mana yang akan dihapus, setelah itu pengguna menekan data tersebut hingga muncul menu untuk menghapus data. Lalu pengguna menekan tombol hapus. Melalui kelas ViewData, data tersebut akan dihapus dari basis data.



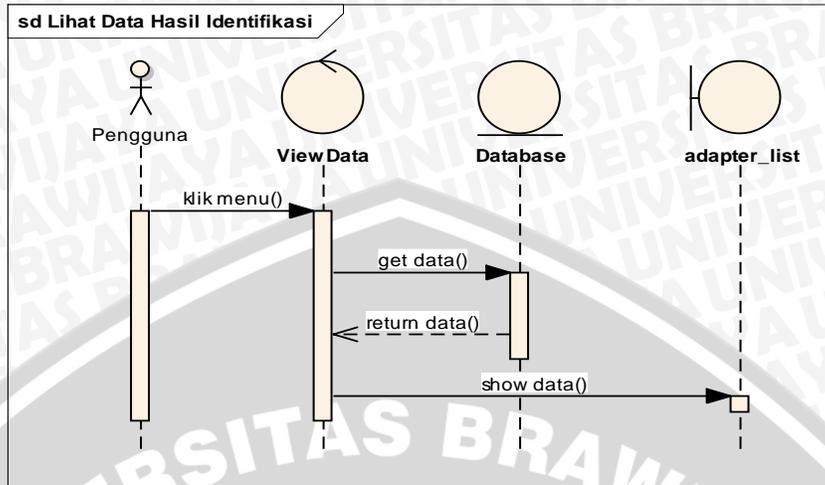
Gambar 4.21. Sequence Diagram Hapus Data Identifikasi

Pada Gambar 4.22 akan menampilkan *sequence diagram* edit data identifikasi. Pertama pengguna masuk ke halaman lihat data, lalu pengguna memilih data mana yang akan diubah, setelah itu pengguna menekan data tersebut hingga muncul menu untuk merubah data. Lalu pengguna menekan tombol edit. Setelah itu pengguna akan diarahkan menuju ke halaman ubah data, setelah selesai merubah data, pengguna akan menekan tombol simpan. Melalui kelas *ViewData*, data tersebut akan disimpan dari basis data.



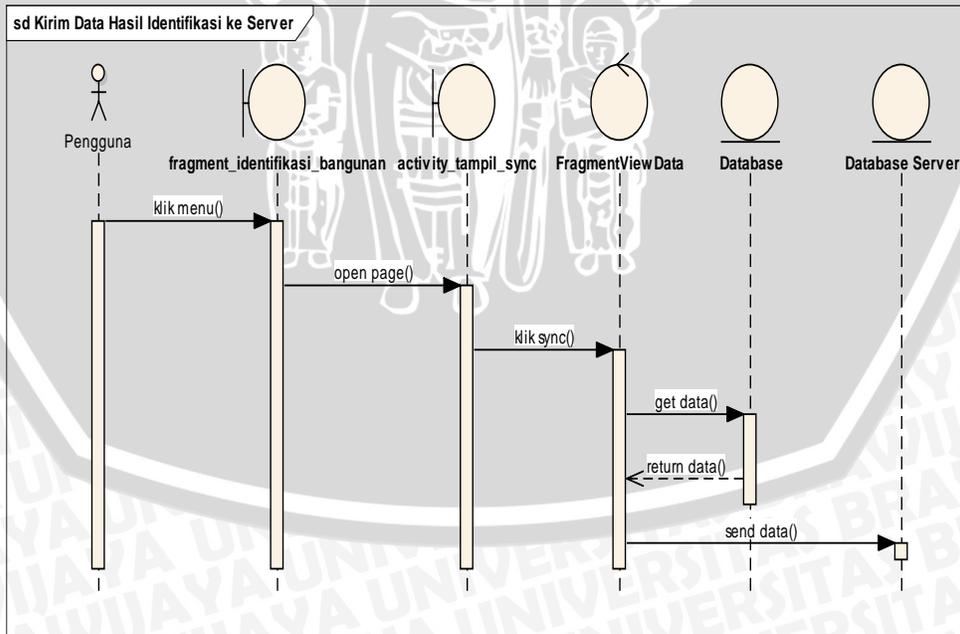
Gambar 4.22. *Sequence Diagram* Edit Data Identifikasi

Pada Gambar 4.23 akan menampilkan *sequence diagram* lihat data identifikasi. Pertama pengguna menekan tombol lihat data, lalu pengguna akan diarahkan menuju ke halaman lihat data, pada halaman tersebut, data akan ditampilkan melalui kelas *ViewData*, yang mana kelas tersebut mengambil data dari basis data pada aplikasi.



Gambar 4.23. Sequence Diagram Lihat Data Identifikasi

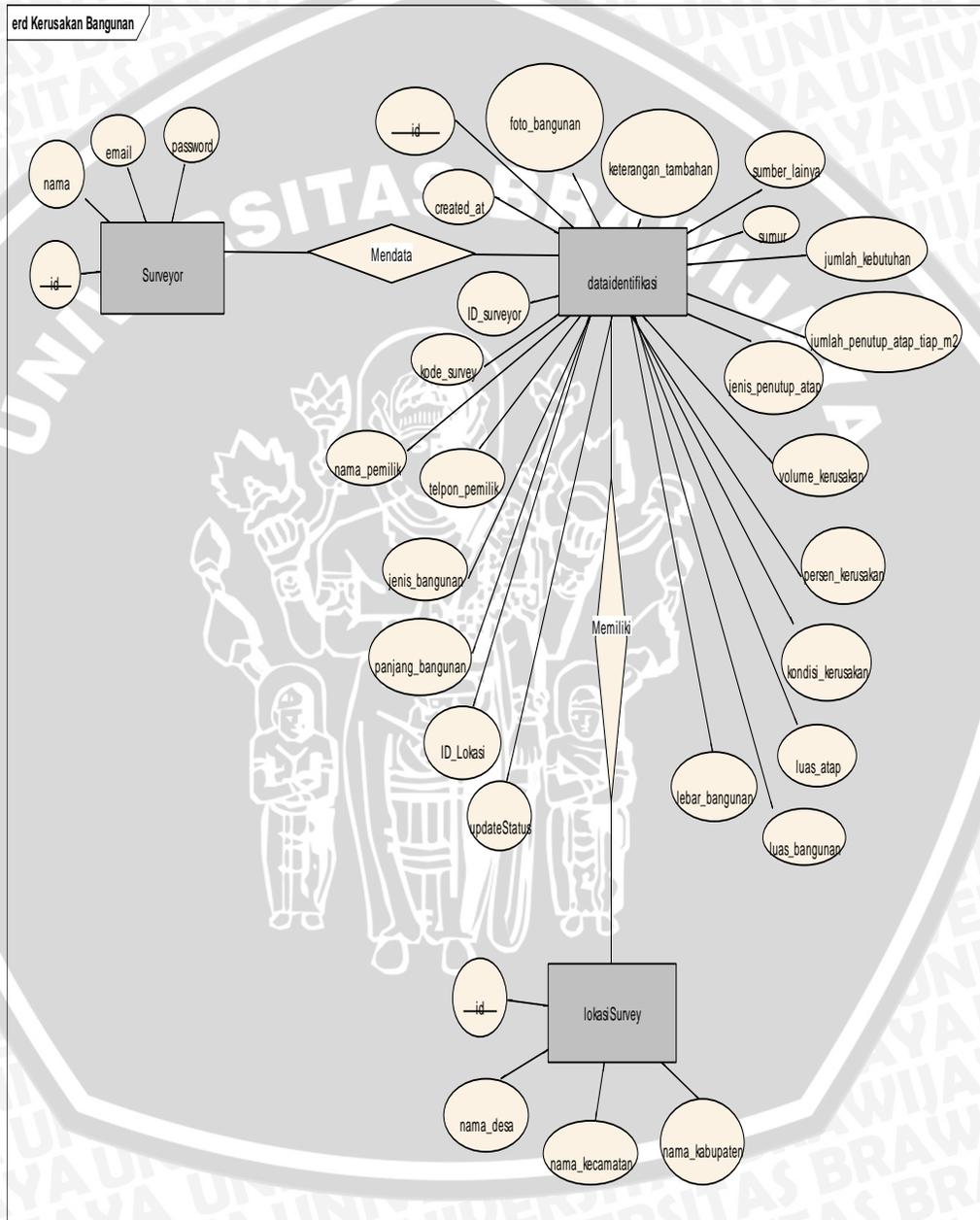
Pada Gambar 4.24 akan menampilkan *sequence diagram* kirim data hasil identifikasi ke server. Pertama pengguna menekan tombol lihat data, lalu pengguna akan diarahkan menuju ke halaman *fragment_identifikasi_bangunan*, pada halaman tersebut, pengguna menggeser halaman tersebut untuk menuju ke halaman *activity_tampil_sync*. Pada halaman tersebut pengguna menekan tombol *sync*. Melalui kelas *fragmentViewData*, data akan diambil dari basis data aplikasi lalu diubah menjadi format *json*, kemudian data tersebut barulah dikirim menuju ke basis data pada server aplikasi.



Gambar 4.24. Sequence Diagram Kirim Data Hasil Identifikasi ke Server

4.2.5. Perancangan Basis Data

Perancangan basis data digunakan untuk mengetahui bagaimana data yang telah dimasukkan ke dalam sistem bisa tersimpan di dalam basis data tersebut. Yang mana data yang tersimpan di dalam basis data meliputi data identifikasi, data identitas surveyor dan data lokasi *survey*. Perancangan basis data tersebut terlihat pada Gambar 4.25.



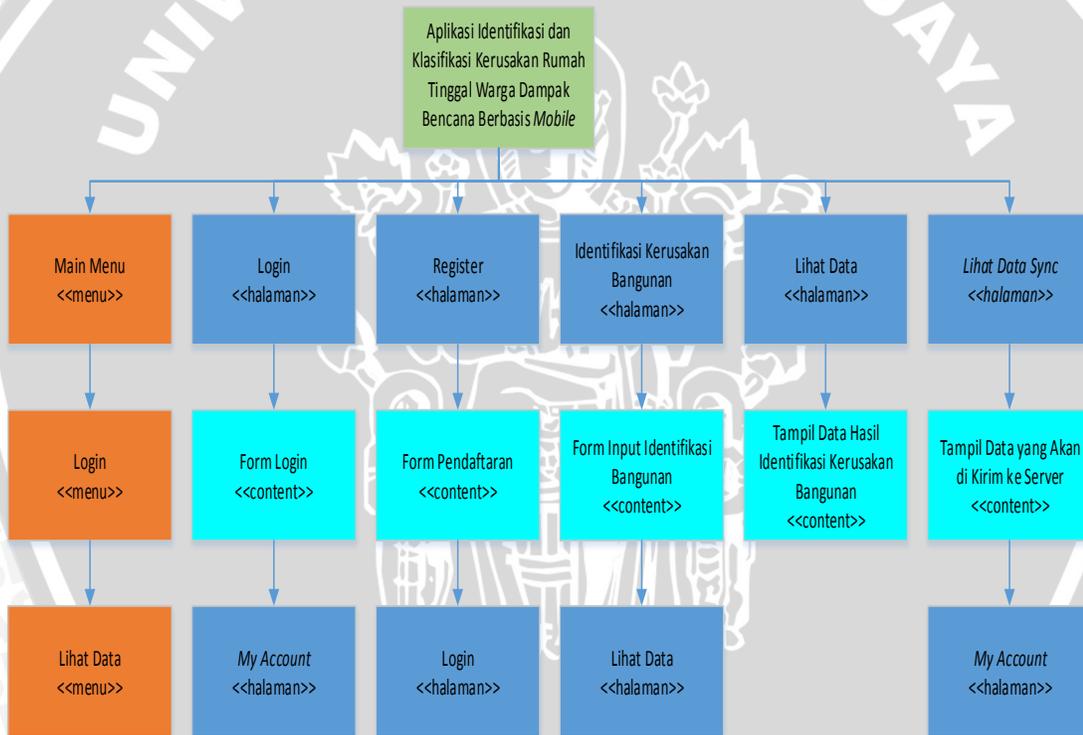
Gambar 4.25. Perancangan Basis Data dataidentifikasi

4.2.6. Perancangan Navigasi dan Antarmuka

Pada bagian ini akan menjelaskan bagaimana rancangan antarmuka dari sistem aplikasi perangkat bergerak identifikasi dan klasifikasi kerusakan rumah warga dampak bencana, serta alur navigasi dari setiap halaman yang ada pada sistem tersebut.

4.2.6.1. Perancangan *Screen Flow*

Di dalam aplikasi identifikasi dan klasifikasi kerusakan rumah warga dampak bencana ini terdapat beberapa halaman seperti halaman utama yang berisi informasi penggunaan aplikasi, halaman tambah data survey, halaman lihat data kebutuhan material dan halaman tentang. Gambaran peta navigasi aplikasi tersebut terlihat seperti pada Gambar 4.27.



Gambar 4.26. Peta Navigasi Aplikasi

Pada Gambar 4.26 terdapat 3 elemen navigasi aplikasi yaitu menu, halaman serta content, yang mana elemen - elemen tersebut dibedakan menurut warnanya masing – masing.

Gambar 4.27 akan menampilkan rancangan *flow screen* dari aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana berbasis perangkat bergerak (*mobile*).



Gambar 4.27. Flow Screen Aplikasi

Berdasarkan peta navigasi yang telah dijelaskan melalui Gambar 4.26, maka akan didapatkan *screen flow* dari aplikasi, yang mana *screen flow* tersebut dijelaskan pada Gambar 4.27. Pada gambar tersebut menampilkan bagaimana aliran tampilan aplikasi ketika memilih menu yang terdapat pada aplikasi tersebut, seperti pada saat memilih menu tambah data, lihat data kebutuhan, *login*, dan *register*. Selain itu juga akan

ditampilkan bagaimana alur ketika kita memilih menu kembali yang terdapat pada aplikasi ataupun pada saat menekan tombol *back* yang terdapat pada perangkat.

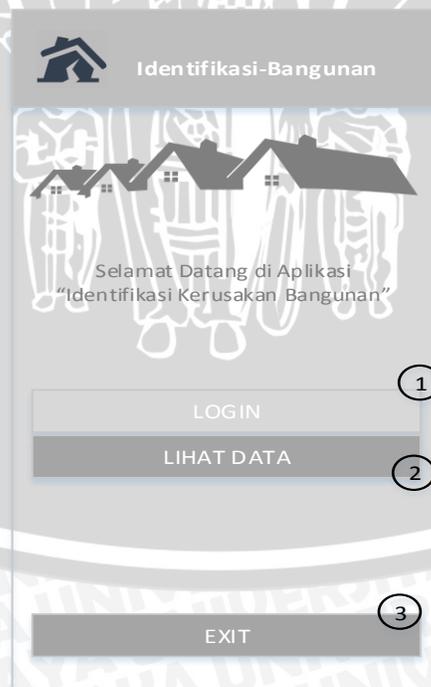
4.2.6.2. Perancangan Antar Muka

Pada bagian ini akan dijelaskan bagaimana perancangan tampilan antar muka dari aplikasi identifikasi dan klasifikasi kerusakan rumah warga dampak bencana ini. Yang mana perancangan tersebut dapat terlihat pada Gambar 4.29 sampai dengan Gambar 4.34.

a. Halaman Utama

Halaman utama merupakan halaman pertama yang akan tampil pada saat pengguna membuka aplikasi. Di dalam halaman utama terdapat tiga tombol utama, yaitu tombol untuk login, lihat data, dan keluar dari aplikasi.

Jika pengguna menekan tombol *login*, maka pengguna akan diarahkan menuju halaman *login*. Selanjutnya jika pengguna memilih menu lihat data, maka pengguna akan diarahkan menuju halaman lihat data kebutuhan. Dan terakhir jika pengguna memilih menu *exit*, maka aplikasi akan menampilkan pemberitahuan apakah pengguna benar – benar ingin keluar dari aplikasi, jika memilih ya, maka aplikasi akan tertutup.



Gambar 4.28. Rancangan Halaman Utama

Pada Gambar 4.28, tiap – tiap bagian antarmuka memiliki keterangan sebagai berikut :

1. Tombol untuk *login*
2. Tombol untuk lihat data
3. Tombol untuk keluar dari aplikasi

b. Halaman *Login*

Halaman *login* merupakan halaman yang berisi form yang harus diisi pengguna jika pengguna tersebut ingin masuk ke dalam sistem aplikasi, yang meliputi alamat email dan *password*.

The image shows a mobile application interface for 'Identifikasi-Bangunan'. At the top, there is a home icon and the text 'Identifikasi-Bangunan'. Below this is a large 'Login' heading. There are two input fields: 'Email' and 'Password'. Below the input fields are two buttons: 'LOGIN' and 'REGISTER'. Three numbered callouts (1, 2, 3) point to the Email field, the LOGIN button, and the REGISTER button respectively.

Gambar 4.29. Rancangan Halaman *Login*

Pada Gambar 4.29, tiap – tiap bagian antarmuka memiliki keterangan sebagai berikut :

1. Form untuk memasukkan alamat email dan *password*
2. Tombol untuk *login* ke dalam sistem
3. Tombol untuk melakukan registrasi atau pendaftaran.

c. Halaman *Register*

Halaman *register* merupakan halaman yang berisi form yang harus diisi pengguna jika pengguna tersebut ingin terdaftar ke dalam

sistem aplikasi, yang meliputi nama lengkap, alamat email dan *password*.

The image shows a mobile application registration screen. At the top, there is a home icon and the text 'Identifikasi - Bangunan'. Below that is the title 'Pendaftaran'. There are three input fields: 'Nama Lengkap', 'Email', and 'Password'. A 'DAFTAR' button is located at the bottom of the form. A circled '1' is next to the 'Nama Lengkap' field, and a circled '2' is next to the 'DAFTAR' button.

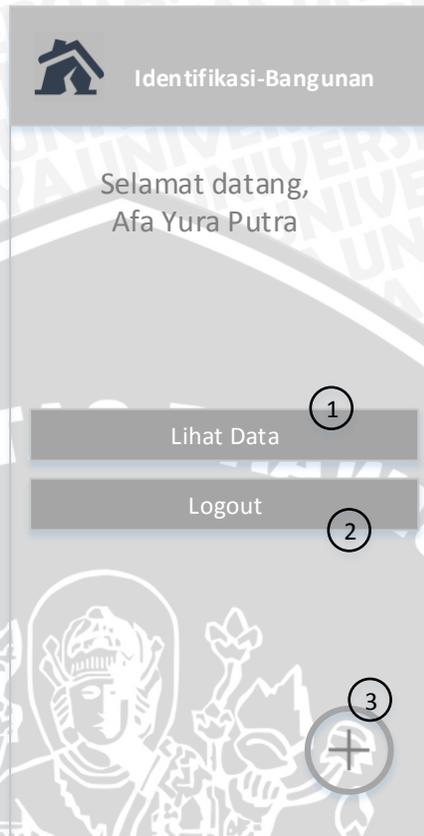
Gambar 4.30. Rancangan Halaman *Register*

Pada Gambar 4.30, tiap – tiap bagian antarmuka memiliki keterangan sebagai berikut :

1. Form untuk memasukkan nama lengkap, alamat email dan *password*
2. Tombol untuk *pendaftaran* ke dalam sistem

d. Halaman *My Account*

Halaman *my account* merupakan halaman yang akan tampil jika pengguna berhasil melakukan *login* ke dalam sistem. Pada halaman tersebut terdapat tiga tombol utama, yaitu tombol lihat data, tombol *logout*, serta tombol tambah data.



Gambar 4.31. Rancangan Halaman My Account

Pada Gambar 4.31, tiap – tiap bagian antarmuka memiliki keterangan sebagai berikut :

1. Tombol untuk melihat data hasil identifikasi
2. Tombol untuk *logout* dari sistem
3. Tombol untuk Menambah data identifikasi

e. Halaman Tambah Data

Halaman tambah data merupakan halaman yang berfungsi untuk menambahkan data identifikasi kerusakan dari bangunan yang akan dilakukan *survey*. Di dalam halaman tambah data terdapat kolom – kolom isian data, yang mana kolom – kolom tersebut terbagi menjadi tiga bagian, yaitu kolom untuk data identitas bangunan, kolom untuk data identifikasi kerusakan bangunan, serta kolom untuk keterangan tambahan.

Gambar 4.32. Rancangan Halaman Tambah Data

Pada Gambar 4.32, tiap – tiap bagian antarmuka memiliki keterangan sebagai berikut :

1. Form untuk input data identitas bangunan
2. Form untuk input data identifikasi kerusakan bangunan
3. Form untuk input data keterangan tambahan
4. Tombol untuk menyimpan hasil masukan data identifikasi bangunan

f. Halaman Lihat Data Identifikasi

Halaman lihat data identifikasi merupakan halaman yang akan menampilkan data hasil identifikasi yang telah dimasukkan ke dalam sistem aplikasi, pada halaman ini terdapat dua tombol, yaitu tombol untuk mengirim data ke server pada komputer, serta tombol untuk kembali ke halaman sebelumnya.

Identifikasi-Bangunan

IDENTIFIKASI BANGUNAN DATA IDENTIFIKASI

DATA IDENTIFIKASI

Kode Survey : AfaSeloBeKeKe ①
Waktu Survey : 14-12-2015 || 04:36:29
Nama Pemilik : Selong
Luas Bangunan : 115.5 m2
Luas Atap : 127 m2
Kondisi Kerusakan : Ringan
Persen Kerusakan : 36 %
Volume Kerusakan : 45.7 m2
Jenis Penutup Atap : Asbes
Jumlah Kebutuhan : 29 Buah
Data Masuk Server : no

② ③

Gambar 4.33. Rancangan Lihat Data Hasil Identifikasi

Tiap – tiap nomor bagian antar muka yang terdapat pada Gambar 4.33. memiliki keterangan sebagai berikut

1. Tampilan data hasil identifikasi kerusakan bangunan
2. Tombol untuk mengirimkan data hasil identifikasi kerusakan bangunan ke server pada komputer
3. Tombol untuk kembali ke halaman sebelumnya

BAB 5 IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi aplikasi perangkat bergerak berdasarkan analisis kebutuhan dan rancangan yang terdapat di dalam Bab IV. Pembahasan pada bab ini terdiri dari penjelasan mengenai spesifikasi perangkat keras dan sistem yang dibutuhkan dalam implementasi. Selain itu juga akan dibahas mengenai apa saja batasan-batasan implementasi, implementasi class dan asset, serta implementasi kode program dari aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana berbasis *mobile*.

5.1. Spesifikasi Sistem

Analisis kebutuhan dan rancangan dari aplikasi perangkat bergerak identifikasi dan klasifikasi kerusakan rumah warga dampak bencana yang terdapat di dalam Bab IV akan menjadi dasar acuan dalam implementasi di bab ini. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1. Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam proses implementasi aplikasi perangkat bergerak identifikasi dan klasifikasi kerusakan rumah warga dampak bencana dapat terlihat di dalam Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
<i>System Model</i>	ASUS A450C Series
<i>Processor</i>	1.8 GHz Core i5 3337U
<i>Memory</i>	8 GB 1600 MHz DDR3
<i>Display</i>	NVIDIA GT 720M

Dalam melakukan proses instalasi dan pengujian, perangkat lunak yang digunakan adalah perangkat bergerak berbasis sistem operasi Android dengan spesifikasi seperti yang terlihat di dalam Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Keras pada Perangkat Bergerak

Nama Komponen	Spesifikasi
<i>System Model</i>	OnePlus One

<i>Processor</i>	Qualcomm Snapdragon 801 Quad-core 2.46 GHz
<i>Memory</i>	3 GB LP-DDR3 1866MHz
<i>Display</i>	Adreno 330 578 MHz

Dalam proses pengembangan aplikasi identifikasi dan klasifikasi kerusakan rumah warga dampak bencana berbasis *mobile* menggunakan perangkat lunak dengan spesifikasi seperti yang terlihat di dalam Tabel 5.3.

Tabel 5.3 Spesifikasi Perangkat Lunak pada Komputer

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 8
Bahasa Pemrograman	Java
IDE (<i>Integrated Development Environment</i>)	Android Studio 1.3.2

Selain itu dalam proses implementasi dan pengujian dari pengembangan aplikasi ini menggunakan perangkat lunak pada perangkat bergerak dengan spesifikasi seperti yang terlihat di dalam Tabel 5.4.

Tabel 5.4 Spesifikasi Perangkat Lunak pada Perangkat Bergerak

Nama Komponen	Spesifikasi
<i>Platform</i>	Android 4.4+

5.2. Batasan – Batasan Implementasi

Pada implementasi aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana berbasis *mobile* ini memiliki batasan – batasan dalam prosesnya, yaitu sebagai berikut :

1. Aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana berbasis *mobile* ini dirancang untuk dapat berjalan pada perangkat bergerak dengan sistem operasi minimum Android 4.4 Kitkat.
2. Aplikasi membutuhkan koneksi internet untuk melakukan proses login dan mengirim data identifikasi ke server, sedangkan untuk melakukan proses identifikasi, aplikasi ini tidak membutuhkan adanya proses koneksi internet.

3. Penyimpanan data yang digunakan pada *local storage* menggunakan *SQLite*.
4. Penyimpanan data yang digunakan pada server menggunakan basis data *MySQL*.

5.3. Implementasi Basis Data

Implementasi penyimpanan data pada aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana berbasis *mobile* menggunakan basis data untuk menyimpan data identitas surveyor, serta data identifikasi kerusakan bangunan. Implementasi dari basis data pada penyimpanan lokal di perangkat bergerak menggunakan *SQLite*, sedangkan untuk penyimpanan data pada server menggunakan basis data *MySQL*. Implementasi basis data berupa struktur tabel yang digunakan dalam pembuatan basis data. Tabel 5.5 merupakan struktur dari tabel data identifikasi pada basis data di penyimpanan lokal perangkat bergerak, Tabel 5.6 merupakan struktur dari tabel data surveyor pada basis data di server komputer, sedangkan Tabel 5.7 merupakan struktur dari tabel identifikasi kerusakan bangunan pada basis data di server komputer. Tabel – tabel tersebut merupakan hasil implementasi dari rancangan basis data yang terdapat pada bab sebelumnya.

Perintah *SQLite* dan *SQL* yang digunakan untuk mengimplementasikan basis data dapat dilihat pada Kode 5.1 sampai Kode 5.3.

```

1. private static final String db_create = "create table "
2.     + TABLE_NAME + "("
3.     + COLUMN_ID + " integer primary key autoincrement, "
4.     + COLUMN_DATE + " DATETIME DEFAULT CURRENT_TIMESTAMP, "
5.     + COLUMN_SURVEYOR + " varchar(50) not null, "
6.     + COLUMN_KODE_SURVEY+ " varchar(50) not null, "
7.     + COLUMN_NAME + " varchar(50) not null, "
8.     + COLUMN_TELPON + " varchar(50) not null, "
9.     + COLUMN_DESA + " varchar(50) not null, "
10.    + COLUMN_KECAMATAN + " varchar(50) not null, "
11.    + COLUMN_KABUPATEN + " varchar(50) not null, "
12.    + COLUMN_JENIS + " varchar(50) not null, "
13.    + COLUMN_PANJANG + " double(50) not null, "
14.    + COLUMN_LEBAR + " double(50) not null, "
15.    + COLUMN_LUAS_BANGUNAN + " double(10) not null, "
16.    + COLUMN_LUAS_ATAP + " double(10) not null, "
17.    + COLUMN_KONDISI_RUSAK + " varchar(50) not null, "
18.    + COLUMN_PERSEN_RUSAK + " varchar(50) not null, "
19.    + COLUMN_VOLUME_RUSAK + " double(50) not null, "
20.    + COLUMN_JENIS_PENUTUP_ATAP + " varchar(50) not null, "
21.    + COLUMN_ASUMSI_JUMLAH + " double(50) not null, "
22.    + COLUMN_JUMLAH_KEBUTUHAN + " varchar(50) not null, "
23.    + COLUMN_SUMBER_AIR + " varchar(50) not null, "
24.    + COLUMN_SUMBER_AIR_LAIN + " varchar(50) not null, "
25.    + COLUMN_KETERANGAN_TAMBAHAN + " varchar(250) not null, "
```

```

26.         + COLUMN_FOTO_BANGUNAN + " BLOB not null, "
27.         + COLUMN_UPDATE_STATUS + " text(50) not null);";

```

Kode 5.1. Perintah SQLite Membuat Tabel data_identifikasi

Pada Kode 5.1 dijelaskan perintah SQLite yang digunakan untuk membuat tabel data_identifikasi di dalam basis data sistem. Terdapat 1 kolom PRIMARY KEY dan 24 kolom biasa.

```

1. CREATE TABLE IF NOT EXISTS `dataidentifikasi` (
2.   `id` int(11) NOT NULL,
3.   `created_at` varchar(50) NOT NULL,
4.   `nama_surveyor` varchar(50) NOT NULL,
5.   `kode_survey` varchar(50) NOT NULL,
6.   `nama_pemilik` varchar(50) NOT NULL,
7.   `telepon_pemilik` varchar(50) NOT NULL,
8.   `nama_desa` varchar(50) NOT NULL,
9.   `nama_kecamatan` varchar(50) NOT NULL,
10.  `nama_kabupaten` varchar(50) NOT NULL,
11.  `jenis_bangunan` varchar(50) NOT NULL,
12.  `panjang_bangunan` double(50,9) NOT NULL,
13.  `lebar_bangunan` double(50,9) NOT NULL,
14.  `luas_bangunan` double(50,9) NOT NULL,
15.  `luas_atap` double(50,9) NOT NULL,
16.  `kondisi_kerusakan` varchar(50) NOT NULL,
17.  `persen_kerusakan` varchar(50) NOT NULL,
18.  `volume_kerusakan` double(50,9) NOT NULL,
19.  `jenis_penutup_atap` varchar(50) NOT NULL,
20.  `jumlah_penutup_atap_tiap_m2` double(50,9) NOT NULL,
21.  `jumlah_kebutuhan` varchar(50) NOT NULL,
22.  `sumur` varchar(50) NOT NULL,
23.  `sumber_lainnya` varchar(50) NOT NULL,
24.  `keterangan_tambahan` varchar(50) NOT NULL,
25.  `foto_bangunan` blob NOT NULL,
26.  PRIMARY KEY (`id`)
27. )

```

Kode 5.2. Perintah SQL Membuat Tabel dataidentifikasi

Pada Kode 5.2 dijelaskan perintah SQL yang digunakan untuk membuat tabel dataidentifikasi di dalam basis data sistem. Terdapat 1 kolom PRIMARY KEY dan 23 kolom biasa.

```

1. CREATE TABLE IF NOT EXISTS `surveyor` (
2.   `id` int(11) NOT NULL AUTO_INCREMENT,
3.   `nama` varchar(50) NOT NULL,
4.   `email` varchar(100) NOT NULL,
5.   `password` varchar(80) NOT NULL,
6.   PRIMARY KEY (`id`),
7.   UNIQUE KEY `email` (`email`)
8. )

```

Kode 5.3. Perintah SQL Membuat Tabel surveyor

Pada Kode 5.3 dijelaskan perintah SQL yang digunakan untuk membuat tabel surveyor di dalam basis data sistem. Terdapat 1 kolom PRIMARY KEY, 1 kolom UNIQUE KEY dan 2 kolom biasa.

Tabel 5.5 Struktur Tabel Identifikasi Kerusakan Bangunan di *Local Storage*

No	Nama Field	Tipe Data	Keterangan
1	_id	Integer	Id utama
2	created_at	DateTime	Tanggal pertama kali data dibuat
3	nama_surveyor	Varchar	Nama dari surveyor
4	kode_survey	Varchar	Kode dari data hasil survey
5	nama_pemilik	Varchar	Nama dari pemilik bangunan
6	telepon_pemilik	Varchar	Nomor telepon pemilik bangunan
7	nama_desa	Varchar	Nama desa dari lokasi bangunan
8	panjang_bangunan	Double	Ukuran panjang bangunan
9	lebar_bangunan	Double	Ukuran lebar bangunan
10	luas_bangunan	Double	Ukuran luas bangunan
11	luas_atap	Double	Ukuran luas atap
12	kondisi_kerusakan	Varchar	Kondisi kerusakan bangunan
13	persen_kerusakan	Varchar	Berapa persen kerusakan bangunan
14	volume_kerusakan	Double	Volume dari kerusakan bangunan
15	jenis_penutup_atap	Varchar	Jenis dari penutup atap
16	jumlah_penutup_atap_tiap_m2	Double	Jumlah dari penutup atap setiap m ²
17	jumlah_kebutuhan	Varchar	Jumlah material yang dibutuhkan
18	sumur	Varchar	Kondisi sumur

19	sumber_lainya	Varchar	Kondisi sumber air lainnya
20	keterangan_tambahan	Varchar	Tambahan keterangan lain
21	foto_bangunan	Blob	Foto dari bangunan yang di survey
22	updateStatus	Text	Status dari terkirimnya data ke server

Tabel 5.6 Struktur Tabel Identitas *Surveyor*

No	Nama Field	Tipe Data	Keterangan
1	id	Integer	Id utama
2	nama	Varchar	Nama dari surveyor
3	email	Varchar	Alamat email dari surveyor
4	password	Varchar	Password login untuk surveyor

Tabel 5.7 Struktur Tabel Lokasi *Survey*

No	Nama Field	Tipe Data	Keterangan
1	id	Integer	Id utama
2	nama_desa	Varchar	Nama desa dari lokasi bangunan
3	nama_kecamatan	Varchar	Nama kecamatan dari lokasi bangunan
4	nama_kabupaten	Varchar	Nama wilayah pemilik bangunan

Tabel 5.8 Struktur Tabel Identifikasi Kerusakan Bangunan di Server

No	Nama Field	Tipe Data	Keterangan
1	id	Integer	Id utama
2	created_at	DateTime	Tanggal pertama kali data dibuat
3	nama_surveyor	Varchar	Nama dari surveyor
4	kode_survey	Varchar	Kode dari data hasil survey

5	nama_pemilik	Varchar	Nama dari pemilik bangunan
6	telepon_pemilik	Varchar	Nomor telepon pemilik bangunan
7	nama_desa	Varchar	Nama desa dari lokasi bangunan
8	nama_kecamatan	Varchar	Nama kecamatan dari lokasi bangunan
9	nama_kabupaten	Varchar	Nama wilayah pemilik bangunan
10	jenis_bangunan	Varchar	Jenis dari bangunan
11	panjang_bangunan	Double	Ukuran panjang bangunan
12	lebar_bangunan	Double	Ukuran lebar bangunan
13	luas_bangunan	Double	Ukuran luas bangunan
14	luas_atap	Double	Ukuran luas atap
15	kondisi_kerusakan	Varchar	Kondisi kerusakan bangunan
16	persen_kerusakan	Varchar	Berapa persen kerusakan bangunan
17	volume_kerusakan	Double	Volume dari kerusakan bangunan
18	jenis_penutup_atap	Varchar	Jenis dari penutup atap
19	jumlah_penutup_atap_tiap_m2	Double	Jumlah dari penutup atap setiap m ²
20	jumlah_kebutuhan	Varchar	Jumlah material yang dibutuhkan
21	Sumur	Varchar	Kondisi sumur
22	sumber_lainya	Varchar	Kondisi sumber air lainnya
23	keterangan_tambahan	Varchar	Tambahan keterangan lain
24	foto_bangunan	Blob	Foto dari bangunan yang di survey

5.4. Implementasi *Class* dan *Assets* pada File Program

Setiap *class* dan *assets* yang telah dirancang pada proses perancangan direalisasikan pada beberapa berkas *file* program. Pada implementasi ini terdiri dari dari *class* dan *assets* untuk aplikasi perangkat bergerak. Berkas *class* dan *assets* yang digunakan dalam implementasi aplikasi perangkat bergerak dapat dilihat pada Tabel 5.8 berikut. Yang mana berkas *class* dan *assets* tersebut terdapat di dalam *package* *afayuraputra.identifikasi_bangunan*. Berkas *.java* merupakan *file activity* yang berisi kode program dan *.xml* merupakan *file layout* dari tiap *activity*

Tabel 5.9 Berkas Kelas dan *Assets Layout* pada Aplikasi Perangkat Bergerak

No	Nama Kelas	Nama <i>Layout</i>
1.	Data.java	-
2.	DBDataSource.java	-
3.	DBHelper.java	-
4.	EditData.java	edit_data.xml
5.	FragmentManagerData.java	activity_tampil_sync.xml
6.	Identifikasi.java	-
7.	IdentifikasiBangunan.java	fragment_identifikasi_bangunan.xml
8.	JsonParser.java	-
9.	Login.java	activity_login.xml
10.	MainActivity.java	activity_main.xml
11.	MyAccount.java	activity_myaccount.xml
12.	Register.java	activity_register.xml
13.	SessionManager.java	-
14.	TabsPagerAdapter.java	-
15.	ViewAdapter.java	adapter_list.xml
16.	ViewData.java	activity_tampil.xml
17.	ViewDataNoLogin.java	activity_tampil_no_login.xml

5.5. Implementasi Kode Program

Pada aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana berbasis *mobile* ini memiliki beberapa fungsi utama yang terdapat dalam beberapa kelas di berkas *.java*. Dalam implementasi kode program ini akan menjelaskan implementasi tiap – tiap kelas ke dalam bentuk suatu kode program, sesuai dengan penjelasan pada Tabel 5.8.

5.5.1. Implementasi Kode Program Aplikasi Perangkat Bergerak

Aplikasi perangkat bergerak ini memiliki beberapa fungsi utama yang ada di beberapa kelas pada *package* *afayuraputra.identifikasi_bangunan*. Pada implementasi ini hanya dicantumkan kode program dari fungsi – fungsi untuk memasukkan data identifikasi dan melihat hasil rekomendasi.

a. Kode Program Memasukkan Data Identifikasi Bangunan

```
1. public class IdentifikasiBangunan extends Fragment implements OnClickListener {
2.     @Override
3.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
4.                             Bundle savedInstanceState) {
5.         View rootView = inflater.inflate(R.layout.fragment_identifikasi_bangunan, container, false);
6.         // instansiasi kelas DBDataSource
7.         dataSource = new DBDataSource(this.getActivity());
8.
9.         //membuat sambungan baru ke database
10.        dataSource.open();
11.        values = dataSource.getAllIdentifikasi();
12.        return rootView;
13.    }
14.
15.    @Override
16.    public void onActivityResult(int requestCode, int resultCode, Intent data) {
17.        b = (Bitmap) data.getExtras().get("data");
18.        fotoView.setImageBitmap(b);
19.    }
20.
21.    public static byte[] getBitmapAsByteArray(Bitmap bitmap) {
22.        ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
23.        bitmap.compress(Bitmap.CompressFormat.JPEG, 100, outputStream);
24.        return outputStream.toByteArray();
25.    }
26.
27.    public String getStringImage(Bitmap bmp){
28.        ByteArrayOutputStream baos = new ByteArrayOutputStream();
29.        bmp.compress(Bitmap.CompressFormat.JPEG, 100, baos);
30.        byte[] imageBytes = baos.toByteArray();
```

```

31.         String encodedImage = Base64.encodeToString(imageBytes, Base64
32.         .DEFAULT);
33.         return encodedImage;
34.     }
35.     @Override
36.     public void onClick(View v) {
37.         //inisialisasi data baru (masih kosong)
38.         Data identifikasi = null;
39.         if (edNama.getText() != null && edTelpon.getText() != null &&
40.         edDesa.getText() != null &&
41.         edKecamatan.getText() != null && edKabupaten.getText()
42.         != null &&
43.         edPanjang.getText() != null && edLebar.getText() != nu
44.         ll && edPersen.getText() != null &&
45.         edSumur.getText() != null && edSumberLain.getText() !=
46.         null && edKeteranganTambahhan.getText() != null) {
47.             /* jika field tidak kosong
48.              * maka masukkan ke dalam data identifikasi*/
49.             session = new SessionManager(getActivity().getApplicationC
50.             ontext());
51.             session.checkLogin();
52.             HashMap<String, String> user = session.getUserDetails();
53.             surveyor = user.get(SessionManager.KEY_NAME);
54.             Bundle bun = new Bundle();
55.             //Data d = Long.parseLong(bun.getLong(id));
56.             //id = bun.getLong("id");
57.             nama = edNama.getText().toString();
58.             telpon = edTelpon.getText().toString();
59.             desa = edDesa.getText().toString();
60.             kecamatan = edKecamatan.getText().toString();
61.             kabupaten = edKabupaten.getText().toString();
62.             jenis = dataSpinner.toString();
63.             panjang = Double.parseDouble(edPanjang.getText().toStrin
64.             g());
65.             lebar = Double.parseDouble(edLebar.getText().toString());
66.
67.             kondisiRusak = radioButton.getText().toString();
68.             persenRusak = Double.parseDouble(edPersen.getText().toStri
69.             ng());
70.             Double LB = panjang*lebar;
71.             luasBangunan = Math.round(LB*100.0)/100.0;
72.             x = 1.1;
73.             Double LA = x*luasBangunan;
74.             luasAtap = Math.round(LA*100.0)/100.0;
75.             Double VR = luasAtap*persenRusak/100;
76.             volumeRusak = Math.round(VR*100.0)/100.0;
77.             jenisPenutupAtap = dataSpinner3.toString();
78.             asumsiJumlah = Double.parseDouble(dataSpinner2.toString())
79.             ;
80.             jk = asumsiJumlah*volumeRusak;
81.             jumlahKebutuhan = (int) Math.round(jk + 0.5);
82.             sumberAir = edSumur.getText().toString();
83.             sumberAirLain = edSumberLain.getText().toString();
84.             keteranganTambahhan = edKeteranganTambahhan.getText().toStri
85.             ng());
86.             fotoBangunan = getStringImage(b);
87.             sur = surveyor.substring(0, 3);
88.             nam = nama.substring(0, 4);

```

```

79.         des = desa.substring(0, 2);
80.         kec = kecamatan.substring(0, 2);
81.         kot = kabupaten.substring(0, 2);
82.         kodeSurvey = sur+nam+des+kec+kot;
83.     }

```

Kode 5.4. Kode Program Kelas IdentifikasiBangunan

Pada Kode 5.4 ditampilkan implementasi kode program dari kelas IdentifikasiBangunan. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan nama kelas IdentifikasiBangunan yang mewarisi sifat dari kelas *Fragment* dan mengimplementasi kelas *OnClickListener* .
2. Baris 5 merupakan fungsi untuk membuat antarmuka dari halaman input data identifikasi.
3. Baris 7 – 12 merupakan fungsi untuk membuat sambungan ke basis data.
4. Baris 17 – 33 merupakan fungsi untuk *convert* gambar hasil masukan.
5. Baris 36 – 83 merupakan fungsi yang dijalankan ketika tombol *submit* di eksekusi.

```

1. public class Login extends Activity {
2.     @Override
3.     protected void onCreate(Bundle savedInstanceState) {
4.         super.onCreate(savedInstanceState);
5.         setContentView(R.layout.activity_login);
6.         session = new SessionManager(getApplicationContext());
7.         Toast.makeText(getApplicationContext(),
8.             "User Login Status: " + session.isLoggedIn(), Toast.LE
9.             NGTH_LONG)
10.            .show();
11.         daftar.setOnClickListener(new OnClickListener()
12.            {
13.                @Override
14.                public void onClick(View arg0) {
15.                    Intent daftar = new Intent(Login.this, Register.class)
16.                ;
17.                    startActivity(daftar);
18.                }
19.            });
20.         login.setOnClickListener(new OnClickListener() {
21.                @Override
22.                public void onClick(View v) {
23.                    url = "http://www.afayuraputra.esy.es/Server/login.php
24.                    ?" + "email="
25.                    + email.getText().toString() + "&password="
26.                    + password.getText().toString();
27.                    if (email.getText().toString().trim().length() > 0
28.                        && password.getText().toString().trim().length
29.                        () > 0)
30.                        {
31.                            new Masuk().execute();

```

```
31.         }
32.         else
33.         {
34.             Toast.makeText(getApplicationContext(), "Username/
password anda masih kosong .. ", Toast.LENGTH_LONG).show();
35.         }
36.     }
37. });
38. }
39. public class Masuk extends AsyncTask<String, String, String>
40. {
41.     ArrayList<HashMap<String, String>> contactList = new ArrayList
<HashMap<String, String>>();
42.     ProgressDialog pDialog;
43.     @Override
44.     protected void onPreExecute() {
45.         // TODO Auto-generated method stub
46.         super.onPreExecute();
47.         pDialog = new ProgressDialog(Login.this);
48.         pDialog.setMessage("Mohon menunggu ...");
49.         pDialog.setIndeterminate(false);
50.         pDialog.setCancelable(true);
51.         pDialog.show();
52.     }
53.     @Override
54.     protected String doInBackground(String... arg0) {
55.         JSONParser jParser = new JSONParser();
56.         JSONObject json = jParser.getJSONFromUrl(url);
57.         try {
58.             success = json.getString("success");
59.             Log.e("error", "nilai sukses=" + success);
60.             JSONArray hasil = json.getJSONArray("login");
61.             if (success.equals("1")) {
62.                 for (int i = 0; i < hasil.length(); i++) {
63.                     JSONObject c = hasil.getJSONObject(i);
64.                     String nama = c.getString("nama").trim();
65.                     String email = c.getString("email").trim();
66.                     session.createLoginSession(nama, email);
67.                     Log.e("ok", " ambil data");
68.                 }
69.             } else {
70.                 Log.e("erro", "tidak bisa ambil data 0");
71.             }
72.         } catch (Exception e) {
73.             // TODO: handle exception
74.             Log.e("erro", "tidak bisa ambil data 1");
75.         }
76.         return null;
77.     }
78.     @Override
79.     protected void onPostExecute(String result) {
80.         // TODO Auto-generated method stub
81.         super.onPostExecute(result);
82.         pDialog.dismiss();
83.         if (success.equals("1")) {
84.             a = new Intent(Login.this, MyAccount.class);
85.             startActivity(a);
86.             finish();
87.         } else {
```

```

88.
89.         Toast.makeText(getApplicationContext(), "Username/pass
word yang anda masukkan salah ..", Toast.LENGTH_LONG).show();
90.     }
91.     }
92. }
93. }

```

Kode 5.5. Kode Program Kelas *Login*

Pada Kode 5.5 ditampilkan implementasi kode program dari kelas *Login*. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan nama kelas *Login* yang mewarisi sifat dari kelas *Activity*.
2. Baris 5 merupakan fungsi untuk membuat antarmuka dari halaman *login*
3. Baris 6 – 9 merupakan fungsi untuk melakukan pengecekan apakah user sudah melakukan *login* atau belum.
4. Baris 10 – 37 merupakan fungsi yang dipanggil ketika tombol *register* dieksekusi.
5. Baris 39 – 93 merupakan fungsi yang dipanggil ketika tombol *login* dieksekusi.

```

1. public class Register extends Activity {
2.
3.     private static String url = "http://www.afayuraputra.esy.es/Server
/register.php";
4.     @Override
5.     protected void onCreate(Bundle savedInstanceState) {
6.         super.onCreate(savedInstanceState);
7.         setContentView(R.layout.activity_register);
8.
9.         daftar.setOnClickListener(new View.OnClickListener() {
10.
11.             @Override
12.             public void onClick(View v) {
13.
14.                 if(!emailValid.validate(email.getText().toString()))
15.                 {
16.                     Toast.makeText(Register.this, "Email yang anda mas
ukkan tidak valid", Toast.LENGTH_LONG).show();
17.                     email.setText("");
18.                 }
19.                 else
20.                 {
21.                     new daftarAku().execute();
22.                 }
23.             }
24.         });
25.     }
26.     public class daftarAku extends AsyncTask<String, String, String>
27.     {
28.         String success;
29.         @Override

```

```

30.         protected void onPreExecute() {
31.             super.onPreExecute();
32.             progressDialog = new ProgressDialog(Register.this);
33.             progressDialog.setMessage("Proses mendaftar...");
34.             progressDialog.setIndeterminate(false);
35.             progressDialog.show();
36.
37.         }
38.         @Override
39.         protected String doInBackground(String... params) {
40.             String strnama = nama.getText().toString();
41.             String stremail = email.getText().toString();
42.             String strpassword = password.getText().toString();
43.             List<NameValuePair> nvp = new ArrayList<NameValuePair>();
44.
45.             nvp.add(new BasicNameValuePair("nama", strnama));
46.             nvp.add(new BasicNameValuePair("email", stremail));
47.             nvp.add(new BasicNameValuePair("password", strpassword));
48.             JSONObject json = jsonParser.makeHttpRequest(url, "POST",
49.                 nvp);
50.             try {
51.                 success = json.getString("success");
52.             } catch (Exception e) {
53.                 Toast.makeText(getApplicationContext(), "Error", Toast
54.                     .LENGTH_LONG).show();
55.             }
56.             return null;
57.         }
58.         protected void onPostExecute(String file_url) {
59.             progressDialog.dismiss();
60.             if (success.equals("1")) {
61.                 Toast.makeText(getApplicationContext(), "Regitrasi suk
62. ses", Toast.LENGTH_LONG).show();
63.             } else {
64.                 Toast.makeText(getApplicationContext(), "Registrasi ga
65. gal", Toast.LENGTH_LONG).show();
66.             }
67.         }
68.     }
69.     public class EmailValidator{
70.         private Pattern pattern;
71.         private Matcher matcher;
72.         private static final String EMAIL_PATTERN =
73.             "^[_A-Za-z0-9-]+(\\.[_A-Za-z0-9-]+)*@[A-Za-z0-
74. 9]+(\\.[A-Za-z0-9]+)*(\\.[A-Za-z]{2,})$";
75.         public EmailValidator()
76.         {
77.             pattern = Pattern.compile(EMAIL_PATTERN);
78.         }
79.         public boolean validate(final String hex)
80.         {
81.             matcher = pattern.matcher(hex);
82.             return matcher.matches();
83.         }
84.     }

```

Kode 5.6. Kode Program Kelas *Register*

Pada Kode 5.6 ditampilkan implementasi kode program dari kelas *Register*. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan nama kelas *Register* yang mewarisi sifat dari kelas *Activity*.
2. Baris 7 merupakan fungsi untuk membuat antarmuka dari halaman *Register*
3. Baris 9 – 64 merupakan fungsi yang dijalankan ketika tombol *submit* dieksekusi.
4. Baris 65 – 82 merupakan fungsi yang digunakan untuk melakukan pengecekan alamat email yang dimasukkan oleh pengguna, apakah telah sesuai dengan format atau belum.

```

1. public class MyAccount extends Activity {
2.     @Override
3.     protected void onCreate(Bundle savedInstanceState) {
4.         super.onCreate(savedInstanceState);
5.         setContentView(R.layout.activity_myaccount);
6.         lihat = (Button) findViewById(R.id.lihatData);
7.         lihat.setOnClickListener(new View.OnClickListener() {
8.             @Override
9.             public void onClick(View arg0) {
10.                 Intent lihat = new Intent(MyAccount.this, ViewData.class);
11.                 startActivity(lihat);
12.             }
13.         });
14.         add = (Button) findViewById(R.id.imageButton);
15.         add.setOnClickListener(new View.OnClickListener() {
16.
17.             @Override
18.             public void onClick(View arg0) {
19.                 // TODO Auto-generated method stub
20.                 Intent keli = new Intent(MyAccount.this, Identifikasi.class);
21.                 startActivity(keli);
22.             }
23.         });
24.         session = new SessionManager(getApplicationContext());
25.         Toast.makeText(getApplicationContext(),
26.             "User Login Status: " + session.isLoggedIn(), Toast.LENGTH_LONG)
27.             .show();
28.         session.checkLogin();
29.         HashMap<String, String> user = session.getUserDetails();
30.         nama = user.get(SessionManager.KEY_NAME);
31.         status = (TextView) findViewById(R.id.status);
32.         status.setText(Html.fromHtml("Selamat Datang,<b> " + nama + "</b>"));
33.         logout = (Button) findViewById(R.id.logout);
34.         logout.setOnClickListener(new View.OnClickListener() {
35.
36.             @Override

```

```

37.         public void onClick(View arg0) {
38.             // TODO Auto-generated method stub
39.             session.logoutUser();
40.             finish();
41.         }
42.     });
43. }
44. @Override
45. public void onBackPressed() { }
46.
47. }

```

Kode 5.7. Kode Program Kelas *MyAccount*

Pada Kode 5.7 ditampilkan implementasi kode program dari kelas *MyAccount*. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan nama kelas *MyAccount* yang mewarisi sifat dari kelas *Activity*.
2. Baris 5 merupakan fungsi untuk membuat antarmuka dari halaman *myaccount*
3. Baris 24 – 32 merupakan fungsi yang dijalankan untuk membuat *session* pada saat pengguna berhasil *login*.
4. Baris 33 – 40 merupakan fungsi yang dipanggil ketika tombol *logout* dieksekusi.

```

1. public class FragmentViewData extends ListFragment {
2.     @Override
3.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
4.                             Bundle savedInstanceState) {
5.         View rootView = inflater.inflate(R.layout.activity_tampil_sync,
6.             container, false);
7.         refresh = (Button) rootView.findViewById(R.id.image_refresh);
8.         refresh.setOnClickListener(new View.OnClickListener(){
9.             @Override
10.            public void onClick(View v) {
11.                syncSQLiteMySQLDB();
12.            }
13.        });
14.         Back = (Button) rootView.findViewById(R.id.back);
15.         Back.setOnClickListener(new View.OnClickListener() {
16.             @Override
17.            public void onClick(View arg0) {
18.                Intent mbalik = new Intent(FragmentViewData.this.getActivity(), MyAccount.class);
19.                startActivity(mbalik);
20.            }
21.        });
22.         dataSource = new DBDataSource(this.getActivity());
23.         // buka kontroller
24.         dataSource.open();

```

```

24.         // ambil semua data identifikasi
25.         values = dataSource.getAllIdentifikasi();
26.         // masukkan data identifikasi ke array adapter
27.         ArrayAdapter<Data> adapter = new ArrayAdapter<Data>(getActivit
y(),
28.                 android.R.layout.simple_list_item_1, values);
29.         // set adapter pada list
30.         setListAdapter(adapter);
31.         return rootView;
32.     }
33.
34.     public void syncSQLiteMySQLDB() {
35.         prgDialog = new ProgressDialog(FragmentViewData.this.getActivi
ty());
36.         prgDialog.setMessage("Synching SQLite Data dengan Remote MySQL
DB. Silahkan menunggu...");
37.         prgDialog.setCancelable(true);
38.         //Create AsyncHttpClient object
39.         AsyncHttpClient client = new AsyncHttpClient();
40.         RequestParams params = new RequestParams();
41.         values = dataSource.getAllIdentifikasi();
42.         if (values.size() != 0) {
43.             if (dataSource.dbSyncCount() != 0) {
44.                 prgDialog.show();
45.                 params.put("usersJSON", dataSource.composeJSONfromSQLi
te());
46.                 Log.i("isi json", String.valueOf(dataSource.composeJSO
NfromSQLite()));
47.                 client.post("http://www.afayuraputra.esy.es/Server/dat
aIdentifikasi/insertidentifikasi.php" ,params ,new AsyncHttpResponseHa
ndler(){
48.                     @Override
49.                     public void onSuccess(int statusCode, Header[] hea
ders, byte[] responseBody) {
50.                         //public void onSuccess(String response){
51.                         String response = new String(responseBody);
52.                         System.out.println(response);
53.                         prgDialog.hide();
54.                         try {
55.                             JSONArray arr = new JSONArray(response);
56.                             System.out.println(arr.length());
57.                             for (int i = 0; i < arr.length(); i++) {
58.                                 JSONObject obj = (JSONObject) arr.get(
i);
59.                                 System.out.println(obj.get("id"));
60.                                 System.out.println(obj.get("status"));
61.                                 dataSource.updateSyncStatus(obj.get("i
d").toString(), obj.get("status").toString());
62.                             }
63.                             Toast.makeText(getActivity().getApplicatio
nContext(), "DB Sync completed!", Toast.LENGTH_LONG).show();
64.                             prgDialog.dismiss();
65.                             Intent ten = new Intent(FragmentViewData.t
his.getActivity(), MyAccount.class);
66.                             startActivity(ten);
67.                         } catch (JSONException e) {
68.                             // TODO Auto-generated catch block

```

```

69.         Toast.makeText(getActivity().getApplicatio
nContext(), "Error Occured [Server's JSON response might be invalid]"
, Toast.LENGTH_LONG).show();
70.         e.printStackTrace();
71.         Log.e("error json", String.valueOf(e));
72.     }
73.     }
74.     @Override
75.     public void onFailure(int statusCode, Header[] hea
ders, byte[] responseBody, Throwable error) {
76.         // TODO Auto-generated method stub
77.         prgDialog.hide();
78.         if (statusCode == 404) {
79.             Toast.makeText(getActivity().getApplicatio
nContext(), "Requested resource not found", Toast.LENGTH_LONG).show();
80.         } else if (statusCode == 500) {
81.             Toast.makeText(getActivity().getApplicatio
nContext(), "Something went wrong at server end", Toast.LENGTH_LONG).s
how();
82.         } else {
83.             Toast.makeText(getActivity().getApplicatio
nContext(), "Unexpected Error occurred! [Most common Error: Device mig
ht not be connected to Internet]", Toast.LENGTH_LONG).show();
84.         }
85.     }
86.     });
87.     } else {
88.         Toast.makeText(getActivity().getApplicationContext(),
"SQLite and Remote MySQL DBs are in Sync!", Toast.LENGTH_LONG).show();
89.     }
90.     } else {
91.         Toast.makeText(getActivity().getApplicationContext(), "No
data in SQLite DB, please do enter User name to perform Sync action",
Toast.LENGTH_LONG).show();
92.     }
93.     }

```

Kode 5.8. Kode Program Kelas *FragmentViewData*

Pada Kode 5.7 ditampilkan implementasi kode program dari kelas *FragmentViewData*. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan nama kelas *FragmentViewData* yang mewarisi sifat dari kelas *ListFragment*.
2. Baris 5 merupakan fungsi untuk membuat antarmuka dari halaman tampil data
3. Baris 34 – 91 merupakan fungsi yang digunakan untuk melakukan sinkronisasi data yang telah dimasukkan pengguna ke basisdata pada perangkat bergerak menuju ke basis data yang terdapat pada server komputer

```
1. public class ViewData extends ListActivity implements OnItemLongClickLis
   tener {
2.
3.     @Override
4.     public void onCreate(Bundle savedInstanceState) {
5.
6.         super.onCreate(savedInstanceState);
7.         setContentView(R.layout.activity_tampil);
8.         imageView = (ImageView) findViewById(R.id.foto);
9.         add = (Button) findViewById(R.id.imageButton);
10.        add.setOnClickListener(new View.OnClickListener() {
11.
12.            @Override
13.
14.                dataSource = new DBDataSource(this);
15.                ProgressDialog prgDialog;
16.                // buka kontroller
17.                dataSource.open();
18.
19.                // ambil semua data identifikasi
20.                values = dataSource.getAllIdentifikasi();
21.                if(values.size()!=0) {
22.                    ArrayAdapter<Data> adapt = new ViewAdapter(this, R.layout.
23. adapter_list, values);
24.                    ListView dataList = (ListView) findViewById(android.R.id.l
25. ist);
26.                    dataList.setAdapter(adapt);
27.                    setListAdapter(adapt);
28.                //apabila ada long click
29.                @Override
30.                public boolean onItemLongClick(final AdapterView<?> adapter, View
31. v, int pos,
32.                    final long id) {
33.
34.                //tampilkan alert dialog
35.                final Dialog dialog = new Dialog(this);
36.                dialog.setContentview(R.layout.dialog_view);
37.                dialog.setTitle("Pilih Aksi");
38.                dialog.show();
39.                final Data d = (Data) getListAdapter().getItem(pos);
40.                editButton = (Button) dialog.findViewById(R.id.button_edit_dat
41. a);
42.                delButton = (Button) dialog.findViewById(R.id.button_delete_da
43. ta);
44.
45.                //apabila tombol edit diklik
46.                editButton.setOnClickListener(
47.                    new OnClickListener() {
48.                        @Override
49.                        public void onClick(View v) {
50.                            // TODO Auto-generated method stub
51.                            switchToEdit(d.getId());
52.                            dialog.dismiss();
53.                        }
54.                    }
55.                );
56.                //apabila tombol delete di klik
57.                delButton.setOnClickListener(
58.                    new OnClickListener() {
```

```

54.         @Override
55.         public void onClick(View v) {
56.             // Delete Data
57.             dataSource.deleteData(d.getId());
58.             dialog.dismiss();
59.             finish();
60.             startActivity(getIntent());
61.         }
62.     }
63. );
64. return true;
65. }

```

Kode 5.9. Kode Program Kelas *ViewData*

Pada Kode 5.9 ditampilkan implementasi kode program dari kelas *ViewData*. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan nama kelas *ViewData* yang mewarisi sifat dari kelas *ListActivity* dan mengimplementasi *OnItemClickListener*.
2. Baris 7 merupakan fungsi untuk membuat antarmuka dari halaman tampil data
3. Baris 14 – 25 merupakan fungsi yang akan mengambil data dari basis data dan menampilkannya ke dalam tampilan *ListView*.
4. Baris 28 – 60 merupakan fungsi yang dipanggil ketika salah satu data yang terdapat di *ListView* di tekan panjang (*OnItemClickListener*)

```

1. public class ViewAdapter extends ArrayAdapter<Data>{
2.     ArrayList<Data> data=new ArrayList<Data>();
3.     public ViewAdapter(Context context, int layoutResourceId, Array
4.         List<Data> data) {
5.         super(context, layoutResourceId, data);
6.         this.layoutResourceId = layoutResourceId;
7.         this.context = context;
8.         this.data = data;
9.
10.        dataSource = new DBDataSource(this.context);
11.        // buka kontroller
12.        dataSource.open();
13.        // ambil semua data identifikasi
14.        values = dataSource.getAllIdentifikasi();
15.
16.    }
17.        byte[] outImage= android.util.Base64.decode(d.getFoto_bangu
18.            nan(), Base64.DEFAULT);
19.        Bitmap theImage =BitmapFactory.decodeByteArray(outImage, 0,
20.            outImage.length);
21.        holder.foto.setImageBitmap(theImage);
22.        holder.tid.setText(Long.toString(id));
23.        holder.tnames.setText(surveyor);
24.        holder.tKode.setText(kodeSurvey);
25.        holder.twaktu.setText(waktu);
26.        holder.tNama.setText(nama);

```

```

25.         holder.tTelpon.setText(telpon);
26.         holder.tJenis.setText(jenis);
27.         holder.tPanjang.setText(Double.toString(panjang));
28.         holder.tLebar.setText(Double.toString(lebar));
29.         holder.tLuasB.setText(Double.toString(luasBangunan));
30.         holder.tLuas.setText(Double.toString(luasAtap));
31.         holder.tKondisi.setText(kondisiRusak);
32.         holder.tPersen.setText(Double.toString(persenRusak));
33.         holder.tVolume.setText(Double.toString(volumeRusak));
34.         holder.tJenisP.setText(jenisP);
35.         holder.tJumlahP.setText(Double.toString(asumsiJumlah));
36.         holder.tJumlahK.setText(Double.toString(jk));
37.         holder.tSumur.setText(sumberAir);
38.         holder.tSumberLain.setText(sumberAirLain);
39.         holder.tKeteranganTambahkan.setText(keteranganTambahkan);
40.         holder.tSync.setText(sync);
41.         return row;
42.     }

```

Kode 5.10. Kode Program Kelas *ViewAdapter*

Pada Kode 5.10 ditampilkan implementasi kode program dari kelas *ViewAdapter*. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan nama kelas *ViewAdapter* yang mewarisi sifat dari kelas *ArrayAdapter*.
2. Baris 9 – 14 merupakan fungsi yang akan mengambil data dari basis data.
3. Baris 17 - 41 merupakan fungsi yang akan menampilkan data yang telah diambil dari basis data.

```

1. public class EditData extends Activity implements OnClickListener{
2.         //buat sambungan baru ke database
3.         dataSource = new DBDataSource(this);
4.         dataSource.open();
5.
6.         //masukkan data-
7.         data identifikasi tersebut ke field editor
8.         txId.append(String.valueOf(id));
9.         edNama.setText(nama);
10.        edTelpon.setText(telpon);
11.        edDesa.setText(desa);
12.        edKecamatan.setText(kecamatan);
13.        edKabupaten.setText(kabupaten);
14.        edPanjang.setText(Double.toString(panjang));
15.        edLebar.setText(Double.toString(lebar));
16.        edPersen.setText(Double.toString(persenRusak));
17.        edSumur.setText(sumberAir);
18.        edSumberLain.setText(sumberAirLain);
19.        edKeteranganTambahkan.setText(keteranganTambahkan);
20.        //Data picture = bun.get(position);
21.        byte[] outImage= Base64.decode(fotoBangunan, Base64.DEFAULT
);
22.        ByteArrayInputStream imageStream = new ByteArrayInputStream
(outImage);

```

```
22.         Bitmap theImage =BitmapFactory.decodeByteArray(outImage, 0,
23.             outImage.length);
24.
25.         //set listener pada tombol
26.         btnSave = (Button) findViewById(R.id.button_save_update);
27.         btnSave.setOnClickListener(this);
28.         btnCancel = (Button) findViewById(R.id.button_cancel_update
29.             );
30.         btnCancel.setOnClickListener(this);
31.         btnFoto = (Button) findViewById(R.id.editFotoBangunan);
32.         btnFoto.setOnClickListener(this);
33.     }
34. @Override
35.     public void onClick(View v) {
36.         // TODO Auto-generated method stub
37.         switch(v.getId())
38.         {
39.             // apabila tombol save diklik (update barang)
40.             case R.id.button_save_update :
41.                 data = new Data();
42.                 session = new SessionManager(getApplicationContext(
43.                     ));
44.                 session.checkLogin();
45.                 HashMap<String, String> user = session.getUserDetail
46.                     s();
47.                 data.setNama_surveyor(user.get(SessionManager.KEY_N
48.                     AME));
49.                 data.setNama_pemilik(edNama.getText().toString());
50.                 data.setTelpon_pemilik(edTelpon.getText().toString(
51.                     ));
52.                 data.setNama_desa(edDesa.getText().toString());
53.                 data.setNama_kecamatan(edKecamatan.getText().toStri
54.                     ng());
55.                 data.setNama_kabupaten(edKabupaten.getText().toStri
56.                     ng());
57.                 data.setJenis_bangunan(dataSpinner.toString());
58.                 data.setPanjang_bangunan(Double.parseDouble(edPanja
59.                     ng.getText().toString()));
60.                 data.setLebar_bangunan(Double.parseDouble(edLebar.g
61.                     etText().toString()));
62.                 data.setLuas_bangunan(panjang * lebar);
63.                 data.setLuas_atap(1.1 * luasBangunan);
64.                 data.setKondisi_kerusakan(radioButton.getText().toS
65.                     tring());
66.                 data.setPersen_kerusakan(Double.parseDouble(edPerse
67.                     n.getText().toString()));
68.                 data.setJenis_penutup_atap(dataSpinner3.toString())
69.                 ;
70.                 data.setJumlah_penutup_atap_tiap_m2(Double.parseDou
71.                     ble(dataSpinner2.toString()));
72.                 data.setVolume_kerusakan(luasAtap * persenRusak / 1
73.                     00);
74.                 jk = asumsiJumlah * volumeRusak;
75.                 data.setJumlah_kebutuhan((int) Math.round(jk + 0.5)
76.                     );
77.                 data.setSumur(edSumur.getText().toString());
```

```

64.         data.setSumber_lainya(edSumberLain.getText().toStri
ng());
65.         data.setKeterangan_tambahan(edKeteranganTambahan.ge
tText().toString());
66.         ByteArrayOutputStream baos = new ByteArrayOutputStream
eam();
67.         BitmapDrawable drawable = (BitmapDrawable) edfotoBa
ngunan.getDrawable();
68.         Bitmap bitmap = drawable.getBitmap();
69.         bitmap.compress(Bitmap.CompressFormat.PNG, 100, bao
s);
70.         byte[] fotoBangunan = baos.toByteArray();
71.         String base64String = Base64.encodeToString(fotoBan
gunan, Base64.DEFAULT);
72.         data.setFoto_bangunan(base64String);
73.         sur = surveyor.substring(0, 3);
74.         nam = nama.substring(0, 4);
75.         des = desa.substring(0, 2);
76.         kec = kecamatan.substring(0, 2);
77.         kot = kabupaten.substring(0, 2);
78.         data.setKode_survey(sur+nam+des+kec+kot);
79.         data.setId(id);
80.         dataSource.updateData(data);
81.         Intent i = new Intent(this, ViewData.class);
82.         startActivity(i);
83.        >EditData.this.finish();
84.         dataSource.close();
85.         break;

```

Kode 5.11. Kode Program Kelas *EditData*

Pada Kode 5.11 ditampilkan implementasi kode program dari kelas *EditData*. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan nama kelas *EditData* yang mewarisi sifat dari kelas *Activity* dan mengimplement *OnClickListener*.
2. Baris 7 – 23 merupakan fungsi yang akan menampilkan di dalam *field editor*.
3. Baris 40 - 85 merupakan fungsi yang digunakan untuk menyimpan data yang telah di edit ke dalam basis data perangkat bergerak.

```

1. public class DBDataSource {
2.
3.     //inisialisasi SQLite Database
4.     private SQLiteDatabase database;
5.
6.     //inisialisasi kelas DBHelper
7.     private DBHelper dbHelper;
8.
9.     //DBHelper diinstantiasi pada constructor
10.    public DBDataSource(Context context)
11.    {
12.        dbHelper = new DBHelper(context);
13.    }
14.

```

```
15. //membuka/membuat sambungan baru ke database
16. public void open() throws SQLException {
17.     database = dbHelper.getWritableDatabase();
18. }
19.
20. //menutup sambungan ke database
21. public void close() {
22.     dbHelper.close();
23. }
24.
25. //method untuk create/insert data ke database
26. public Data createIdentifikasi(String surveyor, String kodeSurvey, String nama, String telpon, String desa, String kecamatan, String kabupaten,
27.     String jenis, Double panjangBangunan, Double lebarBangunan, Double luasBangunan, Double luasAtap, String kondisiRusak, Double persenRusak,
28.     Double volumeRusak, String jenisPenutupAtap, Double asumsiJumlah, Integer jumlahKebutuhan, /*String path,*/
29.     String sumberAir, String sumberAirLain, String keterangan Tambahan, String fotoBangunan) {
30.     ContentValues values = new ContentValues();
31.     SimpleDateFormat dateFormat = new SimpleDateFormat(
32.         "dd-MM-yyyy || HH:mm:ss", Locale.getDefault());
33.     Date date = new Date();
34.     values.put(DBHelper.COLUMN_DATE, dateFormat.format(date));
35.     values.put(DBHelper.COLUMN_SURVEYOR, surveyor);
36.     values.put(DBHelper.COLUMN_KODE_SURVEY, kodeSurvey);
37.     values.put(DBHelper.COLUMN_NAME, nama);
38.     values.put(DBHelper.COLUMN_TELPON, telpon);
39.     values.put(DBHelper.COLUMN_DESA, desa);
40.     values.put(DBHelper.COLUMN_KECAMATAN, kecamatan);
41.     values.put(DBHelper.COLUMN_KABUPATEN, kabupaten);
42.     values.put(DBHelper.COLUMN_JENIS, jenis);
43.     values.put(DBHelper.COLUMN_PANJANG, panjangBangunan);
44.     values.put(DBHelper.COLUMN_LEBAR, lebarBangunan);
45.     values.put(DBHelper.COLUMN_LUAS_BANGUNAN, luasBangunan);
46.     values.put(DBHelper.COLUMN_LUAS_ATAP, luasAtap);
47.     values.put(DBHelper.COLUMN_KONDISI_RUSAK, kondisiRusak);
48.     values.put(DBHelper.COLUMN_PERSEN_RUSAK, persenRusak);
49.     values.put(DBHelper.COLUMN_VOLUME_RUSAK, volumeRusak);
50.     values.put(DBHelper.COLUMN_JENIS_PENUTUP_ATAP, jenisPenutup
    Atap);
51.     values.put(DBHelper.COLUMN_ASUMSI_JUMLAH, asumsiJumlah);
52.     values.put(DBHelper.COLUMN_JUMLAH_KEBUTUHAN, jumlahKebutuhan);
53.     values.put(DBHelper.COLUMN_SUMBER_AIR, sumberAir);
54.     values.put(DBHelper.COLUMN_SUMBER_AIR_LAIN, sumberAirLain);
55.     values.put(DBHelper.COLUMN_KETERANGAN_TAMBAHAN, keteranganTambahan);
56.     values.put(DBHelper.COLUMN_FOTO_BANGUNAN, fotoBangunan);
57.     values.put(DBHelper.COLUMN_UPDATE_STATUS, "no");
58.
59.     // mengeksekusi perintah SQL insert data
60.     // yang akan mengembalikan sebuah insert ID
```

```
61.         long insertId = database.insert(DBHelper.TABLE_NAME, null,
62.             values);
63.         Cursor cursor = database.query(DBHelper.TABLE_NAME,
64.             allColumns, DBHelper.COLUMN_ID + " = " + insertId,
65.             null,
66.             null, null, null);
67.         // pindah ke data paling pertama
68.         cursor.moveToFirst();
69.
70.
71.         // mengubah objek pada kursor pertama tadi
72.         // ke dalam objek identifikasi
73.         Data newIdentifikasi = cursorToIdentifikasi(cursor);
74.
75.         // close cursor
76.         cursor.close();
77.
78.         // mengembalikan identifikasi baru
79.         return newIdentifikasi;
80.     }
81.
82.     private Data cursorToIdentifikasi(Cursor cursor)
83.     {
84.         // buat objek identifikasi baru
85.         Data identifikasi = new Data();
86.         // debug LOGCAT
87.         Log.v("info", "The getLONG " + cursor.getLong(0));
88.         Log.v("info", "The setLatLng " + cursor.getString(1) + ", "
89.             + cursor.getString(2));
90.
91.         /* Set atribut pada objek identifikasi dengan
92.          * data kursor yang diambil dari database*/
93.         identifikasi.setId(cursor.getLong(0));
94.         identifikasi.setCreated(cursor.getString(1));
95.         identifikasi.setNama_surveyor(cursor.getString(2));
96.         identifikasi.setKode_survey(cursor.getString(3));
97.         identifikasi.setNama_pemilik(cursor.getString(4));
98.         identifikasi.setTelpon_pemilik(cursor.getString(5));
99.         identifikasi.setNama_desa(cursor.getString(6));
100.        identifikasi.setNama_kecamatan(cursor.getString(7));
101.        identifikasi.setNama_kabupaten(cursor.getString(8));
102.
103.        identifikasi.setJenis_bangunan(cursor.getString(9));
104.
105.        identifikasi.setPanjang_bangunan(cursor.getDouble(10)
106.    );
107.        identifikasi.setLebar_bangunan(cursor.getDouble(11));
108.
109.        identifikasi.setLuas_bangunan(cursor.getDouble(12));
110.
111.        identifikasi.setLuas_atap(cursor.getDouble(13));
112.        identifikasi.setKondisi_kerusakan(cursor.getString(14)
113.    );
114.        identifikasi.setPersen_kerusakan(cursor.getDouble(15)
115.    );
116.        identifikasi.setVolume_kerusakan(cursor.getDouble(16)
117.    );
```

```
109.         identifikasi.setJenis_penutup_atap(cursor.getString(1
110.         7));
110.         identifikasi.setJumlah_penutup_atap_tiap_m2(cursor.ge
111.         tDouble(18));
111.         identifikasi.setJumlah_kebutuhan(cursor.getInt(19));
112.         identifikasi.setSumur(cursor.getString(20));
113.         identifikasi.setSumber_lainya(cursor.getString(21));
114.         identifikasi.setKeterangan_tambahan(cursor.getString(
115.         22));
115.         identifikasi.setFoto_bangunan(cursor.getString(23));
116.         identifikasi.setUpdateStatus(cursor.getString(24));
117.
118.         //kembalikan sebagai objek identifikasi
119.         return identifikasi;
120.     }
121.
122.     //mengambil semua data identifikasi
123.     public ArrayList<Data> getAllIdentifikasi() {
124.         ArrayList<Data> dataIdentifikasi = new ArrayList<Data
125.         >();
126.         // select all SQL query
127.         Cursor cursor = database.query(DBHelper.TABLE_NAME,
128.         allColumns, "updateStatus = '"+no+"' ", null
129.         , null, null, null, null);
130.         // pindah ke data paling pertama
131.         cursor.moveToFirst();
132.         // jika masih ada data, masukkan data ke
133.         // data Identifikasi
134.         while (!cursor.isAfterLast()) {
135.             Data identifikasi = cursorToIdentifikasi(cursor);
136.             dataIdentifikasi.add(identifikasi);
137.             cursor.moveToNext();
138.         }
139.         // Make sure to close the cursor
140.         cursor.close();
141.         return dataIdentifikasi;
142.     }
143.
144.     public String composeJSONfromSQLite(){
145.         ArrayList<Data> dataIdentifikasi = new ArrayList<Data
146.         >();
146.         Cursor cursor = database.query(DBHelper.TABLE_NAME, a
147.         llColumns, "updateStatus = '"+no+"' ", null, null, null, null);
148.         cursor.moveToFirst();
149.         // jika masih ada data, masukkan data ke
150.         // data Identifikasi
151.         while (!cursor.isAfterLast()) {
152.             Data identifikasi = cursorToIdentifikasi(cursor);
153.             dataIdentifikasi.add(identifikasi);
154.             cursor.moveToNext();
155.         }
155.         // Make sure to close the cursor
```

```

156.         cursor.close();
157.         Gson gson = new GsonBuilder().create();
158.         return gson.toJson(dataIdentifikasi);
159.     }
160.
161.     public String getSyncStatus(){
162.         String msg = null;
163.         if(this.dbSyncCount() == 0){
164.             msg = "SQLite and Remote MySQL DBs are in Sync!";
165.         }else{
166.             msg = "DB Sync needed";
167.         }
168.         return msg;
169.     }
170.
171.     public int dbSyncCount(){
172.         int count = 0;
173.         Cursor cursor = database.query(DBHelper.TABLE_NAME, a
174.             llColumns, "updateStatus = '"+no+"' ", null, null, null, null);
175.         count = cursor.getCount();
176.         cursor.close();
177.         return count;
178.     }
179.     public void updateSyncStatus(String id, String status){
180.         String updateQuery = "Update data_identifikasi set up
181.         dateStatus = '"+ status +"' where _Id='"+id+"'";
182.         String strFilter = "_id=" + id;
183.         Log.d("query",updateQuery);
184.         database.execSQL(updateQuery);
185.         database.close();
186.     }

```

Kode 5.12. Kode Program Kelas *DBDataSource*

Pada Kode 5.12 ditampilkan implementasi kode program dari kelas *DBDataSource*. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan nama kelas *DBDataSource*.
2. Baris 26 – 79 merupakan fungsi yang digunakan untuk memasukkan data ke dalam basis data.
3. Baris 144 – 176 merupakan fungsi yang digunakan untuk mengirimkan data yang terdapat di basis data perangkat bergerak menuju basis data pada server komputer.
4. Baris 179 – 184 merupakan fungsi yang digunakan untuk mengubah status di basis data bahwa data telah terkirim ke server komputer.

5.5.2. Implementasi Kode Program *Web Service*

Web service ini memiliki fungsi utama yaitu untuk menerima data yang dikirim dari basis data perangkat bergerak dalam format *json*. Pada

implementasi ini hanya dicantumkan kode program dari fungsi – fungsi untuk mendapatkan data identifikasi.

```

1. class DB_Functions {
2.
3.     private $db;
4.
5.     // constructor
6.     function __construct() {
7.         include_once 'db_connect.php';
8.         // connecting to database
9.         $this->db = new DB_Connect();
10.        $this->db->connect();
11.    }
12.
13.    // destructor
14.    function __destruct() {
15.
16.    }
17.
18.    public function storeIdentifikasi($id,$created_at,$nama_surveyor,$
kode_survey,$nama_pemilik,$telepon_pemilik,$nama_desa,$nama_kecamatan,$
nama_kabupaten,$jenis_bangunan,$panjang_bangunan,$lebar_bangunan,$luas
s_bangunan,$luas_atap,$kondisi_kerusakan,$persen_kerusakan,$volume_ker
usakan,$jenis_penutup_atap,$jumlah_penutup_atap_tiap_m2,$jumlah_kebutu
han,$sumur,$sumber_lainya,$keterangan_tambahan,$foto_bangunan) {
19.        // Insert user into database
20.        $result = mysql_query("INSERT INTO dataidentifikasi VALUES($id
,'$created_at','$nama_surveyor','$kode_survey','$nama_pemilik','$telep
on_pemilik','$nama_desa','$nama_kecamatan','$nama_kabupaten','$jenis_b
angunan','$panjang_bangunan','$lebar_bangunan','$luas_bangunan','$luas
_atap','$kondisi_kerusakan','$persen_kerusakan','$volume_kerusakan','$
jenis_penutup_atap','$jumlah_penutup_atap_tiap_m2','$jumlah_kebutuhan'
,'$sumur','$sumber_lainya','$keterangan_tambahan','$foto_bangunan')");
21.
22.        if ($result) {
23.            return true;
24.        } else {
25.            if( mysql_errno() == 1062) {
26.                // Duplicate key - Primary Key Violation
27.                return true;
28.            } else {
29.                // For other errors
30.                return false;
31.            }
32.        }
33.    /**
34.     * Getting all data
35.     */
36.    public function getAllIdentifikasi() {
37.        $result = mysql_query("select * FROM dataidentifikasi");
38.        return $result;
39.    }
40. }

```

Kode 5.13. Kode Program Kelas *DB_Functions*

Pada Kode 5.13 ditampilkan implementasi kode program dari kelas *DB_Functions*. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan nama kelas *DB_Functions* .
2. Baris 3 merupakan deklarasi variabel yang digunakan.
3. Baris 6 - 10 merupakan fungsi untuk menghubungkan ke basis data pada server komputer
4. Baris 18 – 33 merupakan fungsi yang digunakan untuk memasukkan data ke dalam basis data pada server komputer.
5. Baris 37 - 40 merupakan fungsi yang digunakan untuk mengambil data yang ada di dalam basis data server komputer.

```

1. include_once 'db_functions.php';
2.
3. //Create Object for DB_Functions clas
4. $db = new DB_Functions();
5.
6. //Get JSON posted by Android Application
7. $json = $_POST["usersJSON"];
8.
9. //Remove slashes
10. if (get_magic_quotes_gpc()){
11. $json = stripslashes($json);
12. }
13.
14. $data = json_decode($json, true);
15. $a=array();
16. $b=array();
17.
18. for($i=0; $i<count($data) ; $i++)
19. {
20.
21. $res = $db-
    >storeIdentifikasi($data[$i]['id'],$data[$i]['created_at'],$data[$i]['
    nama_surveyor'],$data[$i]['kode_survey'],$data[$i]['nama_pemilik'],$da
    ta[$i]['telpon_pemilik'],$data[$i]['nama_desa'],$data[$i]['nama_kecama
    tan'],$data[$i]['nama_kabupaten'],$data[$i]['jenis_bangunan'],$data[$i
    ]['panjang_bangunan'],$data[$i]['lebar_bangunan'],$data[$i]['luas_bang
    unan'],$data[$i]['luas_atap'],$data[$i]['kondisi_kerusakan'],$data[$i]
    ['persen_kerusakan'],$data[$i]['volume_kerusakan'],$data[$i]['jenis_pe
    nutup_atap'],$data[$i]['jumlah_penutup_atap_tiap_m2'],$data[$i]['jumla
    h_kebutuhan'],$data[$i]['sumur'],$data[$i]['sumber_lainya'],$data[$i][
    'keterangan_tambahan'],$data[$i]['foto_bangunan']);
22.
23.     if($res){
24.         $b["id"] = $data[$i]['id'];
25.         $b["status"] = 'yes';
26.         array_push($a,$b);
27.     }
28.
29.     else{
30.         $b["id"] = $data[$i]['id'];
31.         $b["status"] = 'no';

```

```

32.         array_push($a,$b);
33.
34.     }
35.
36. }
37.
38. echo json_encode($a);

```

Kode 5.14. Kode Program *insertidentifikasi*

Pada Kode 5.14 ditampilkan implementasi kode program *insertidentifikasi*. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan deklarasi untuk memanggil file *db_functions.php*.
2. Baris 5 merupakan deklarasi variabel untuk mendapatkan data *json* yang dikirimkan oleh aplikasi pada perangkat bergerak.
3. Baris 12 merupakan fungsi untuk merubah data *json* menjadi data *array*
4. Baris 22 - 33 merupakan fungsi yang digunakan untuk mengirimkan respon kepada aplikasi perangkat bergerak mengenai data *json* yang telah dikirimkan ke basis data di server komputer.

5.5.3. Implementasi Kode Program Untuk Perhitungan Data Kebutuhan Material

Perhitungan digunakan untuk mengetahui jumlah material yang dibutuhkan berdasarkan data identifikasi yang telah dimasukkan sebelumnya. Pada implementasi ini dicantumkan kode program perhitungan data kebutuhan material dalam format *.php*

```

1. panjang = Double.parseDouble(edPanjang.getText().toString());
2. lebar = Double.parseDouble(edLebar.getText().toString());
3. persenRusak = Double.parseDouble(edPersen.getText().toString());
4. Double LB = panjang*lebar;
5. luasBangunan = Math.round(LB*100.0)/100.0;
6. x = 1.1;
7. Double LA = x*luasBangunan;
8. luasAtap = Math.round(LA*100.0)/100.0;
9. Double VR = luasAtap*persenRusak/100;
10. volumeRusak = Math.round(VR*100.0)/100.0;
11. jenisPenutupAtap = dataSpinner3.toString();
12. asumsiJumlah = Double.parseDouble(dataSpinner2.toString());
13. jk = asumsiJumlah*volumeRusak;
14. jumlahKebutuhan = (int) Math.round(jk + 0.5);

```

Kode 5.15. Kode Program Perhitungan Data Kebutuhan Material

Pada Kode 5.15 ditampilkan implementasi kode program *perhitungan data kebutuhan material*. Berikut penjelasan implementasi kode program :

1. Baris 1, merupakan deklarasi untuk mendapatkan data panjang bangunan
2. Baris 2 merupakan deklarasi untuk mendapatkan data lebar bangunan.
3. Baris 3 merupakan deklarasi untuk mendapatkan data persen kerusakan

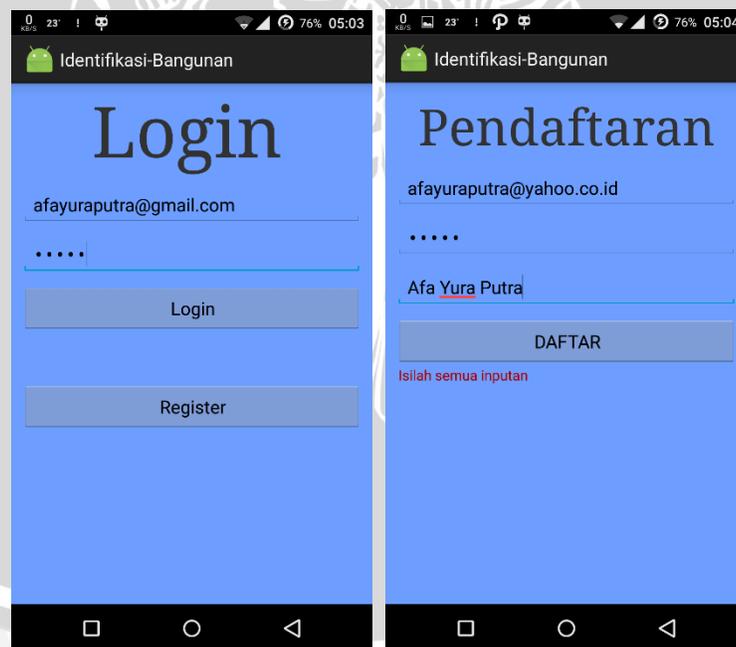
4. Baris 4 merupakan deklarasi untuk menghitung luas bangunan
5. Baris 7 merupakan deklarasi untuk menghitung luas atap bangunan
6. Baris 9 merupakan deklarasi untuk menghitung volume kerusakan bangunan
7. Baris 11 merupakan deklarasi untuk mendapatkan jenis penutup atap bangunan
8. Baris 12 merupakan deklarasi untuk mendapatkan asumsi jumlah penutup atap tiap m² berdasarkan jenis penutup atap yang dipilih
9. Baris 13 merupakan deklarasi untuk menghitung jumlah material yang dibutuhkan.

5.6. Implementasi Antarmuka

Implementasi antarmuka akan menampilkan hasil implementasi dari perancangan antarmuka yang terdapat pada bab sebelumnya. Penjelasan tampilan implementasi antarmuka aplikasi perangkat bergerak yang dibangun dapat dilihat pada Gambar 5.1 sampai Gambar 5.7 berikut.

a. Halaman *Login* dan *Register*

Implementasi dari halaman *Login* dan *Register* dapat dilihat pada Gambar 5.1.



Gambar 5.1. Hasil Implementasi Halaman *Login* dan *Register*

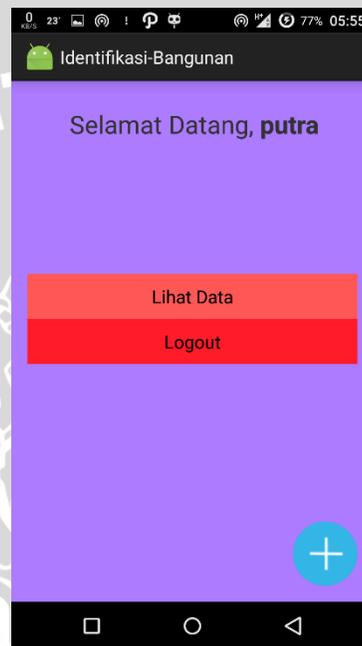
Halaman *Login* merupakan halaman yang digunakan untuk masuk ke dalam sistem pada aplikasi, yaitu dengan cara memasukkan alamat

email serta *password* pada *form* yang terdapat pada halaman tersebut.

Sedangkan Halaman *Register* digunakan untuk melakukan pendaftaran bagi pengguna yang ingin masuk ke dalam sistem aplikasi namun masih belum memiliki akun.

b. Halaman *My Account*

Implementasi halaman *My Account* dapat dilihat pada Gambar 5.2.

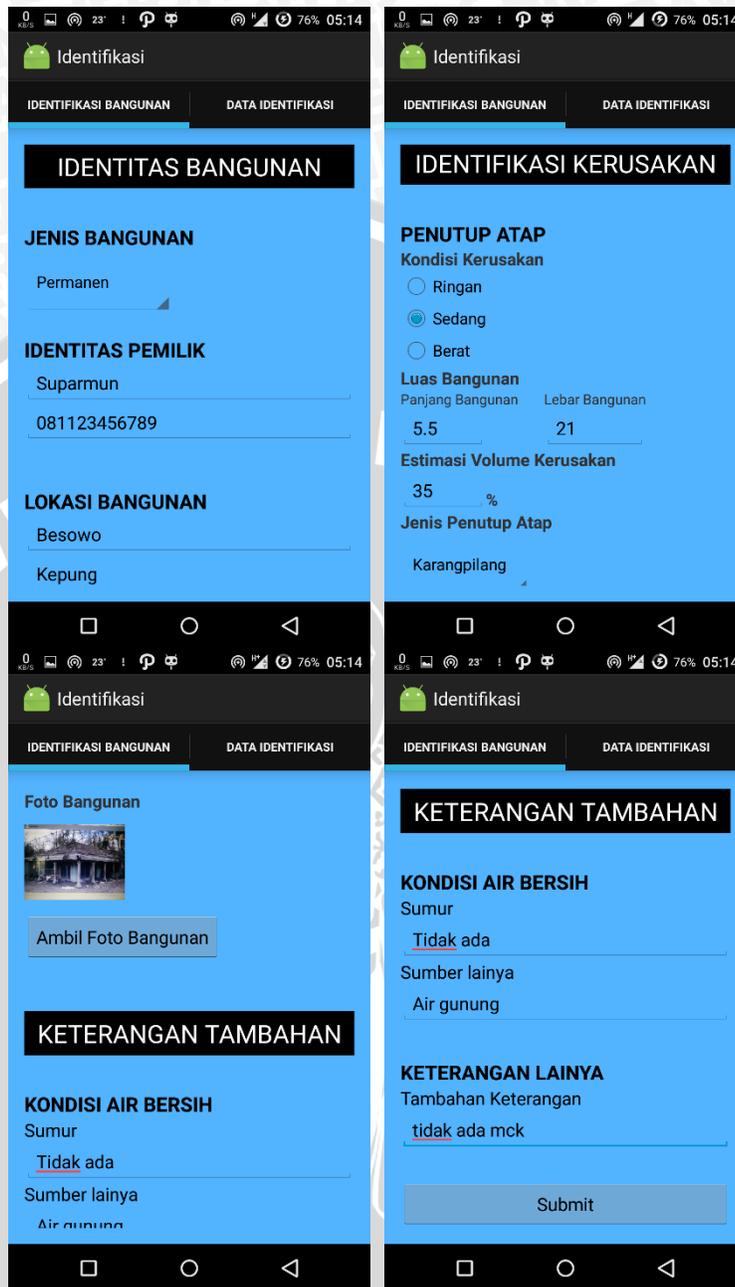


Gambar 5.2. Hasil Implementasi Halaman *My Account*

My Account merupakan halaman yang ditampilkan pertamakali pada saat pengguna berhasil melakukan *Login*.

c. Halaman Identifikasi Bangunan

Identifikasi Bangunan merupakan halaman yang digunakan untuk memasukkan data identifikasi kerusakan bangunan, yang mana pada halaman tersebut terdiri dari 3 bagian yaitu bagian identitas bangunan, identifikasi kerusakan, dan keterangan tambahan.

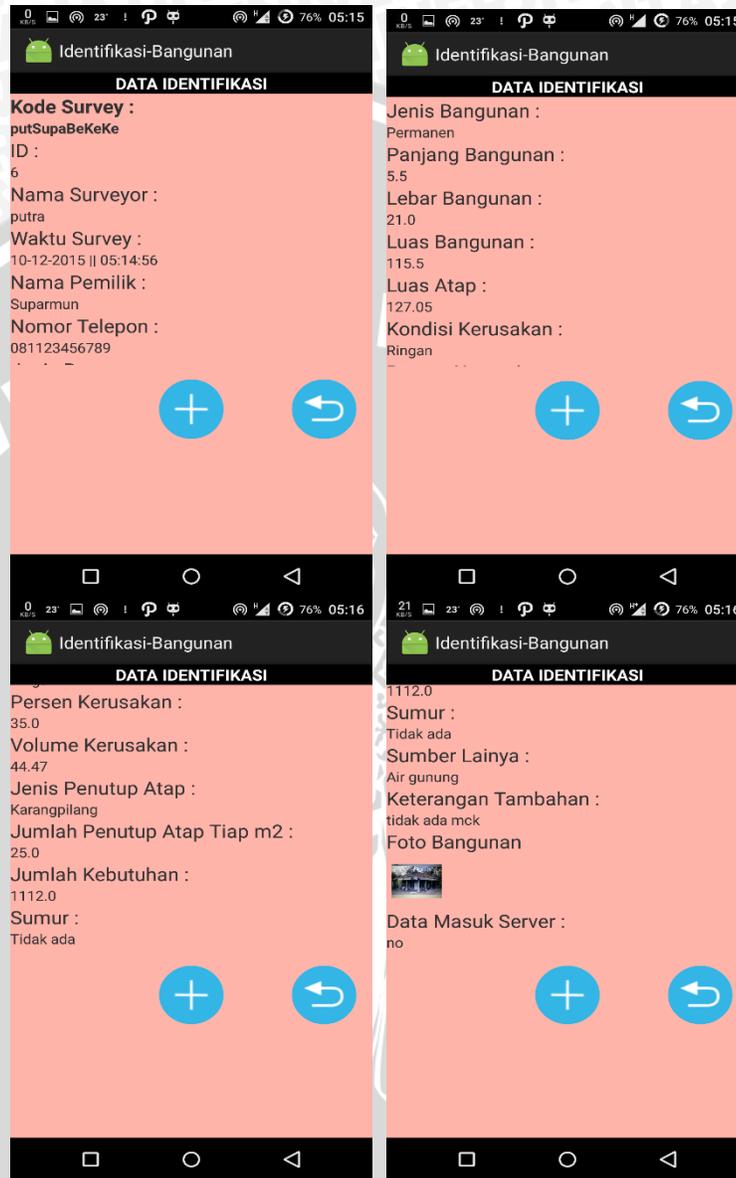


Gambar 5.3. Hasil Implementasi Halaman Identifikasi Bangunan

d. Halaman View Data

View Data merupakan halaman yang digunakan untuk melihat data hasil identifikasi kerusakan bangunan yang telah di *input* ke dalam sistem aplikasi. Pada halaman ini selain menampilkan data yang telah di *input* tersebut, juga akan ditampilkan data seperti kode *survey* yang terbentuk dari data nama *surveyor*, data nama pemilik bangunan, dan data lokasi bangunan; data Luas bangunan, luas atap bangunan,

volume kerusakan bangunan, serta yang data paling utama yaitu data jumlah kebutuhan material.

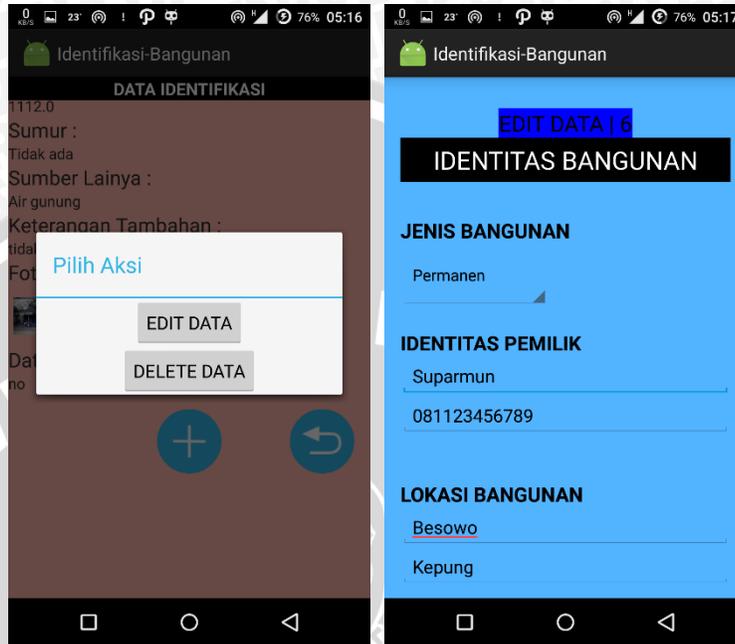


Gambar 5.4. Hasil Implementasi Halaman View Data

e. Halaman *Edit Data*

Edit data merupakan halaman yang digunakan untuk melakukan perubahan pada data yang telah di *input* sebelumnya. Untuk melakukan perubahan pada data, pengguna harus memilih terlebih dahulu memilih data mana yang akan dilakukan perubahan, setelah itu pengguna akan menekan lama data tersebut sampai muncul jendela

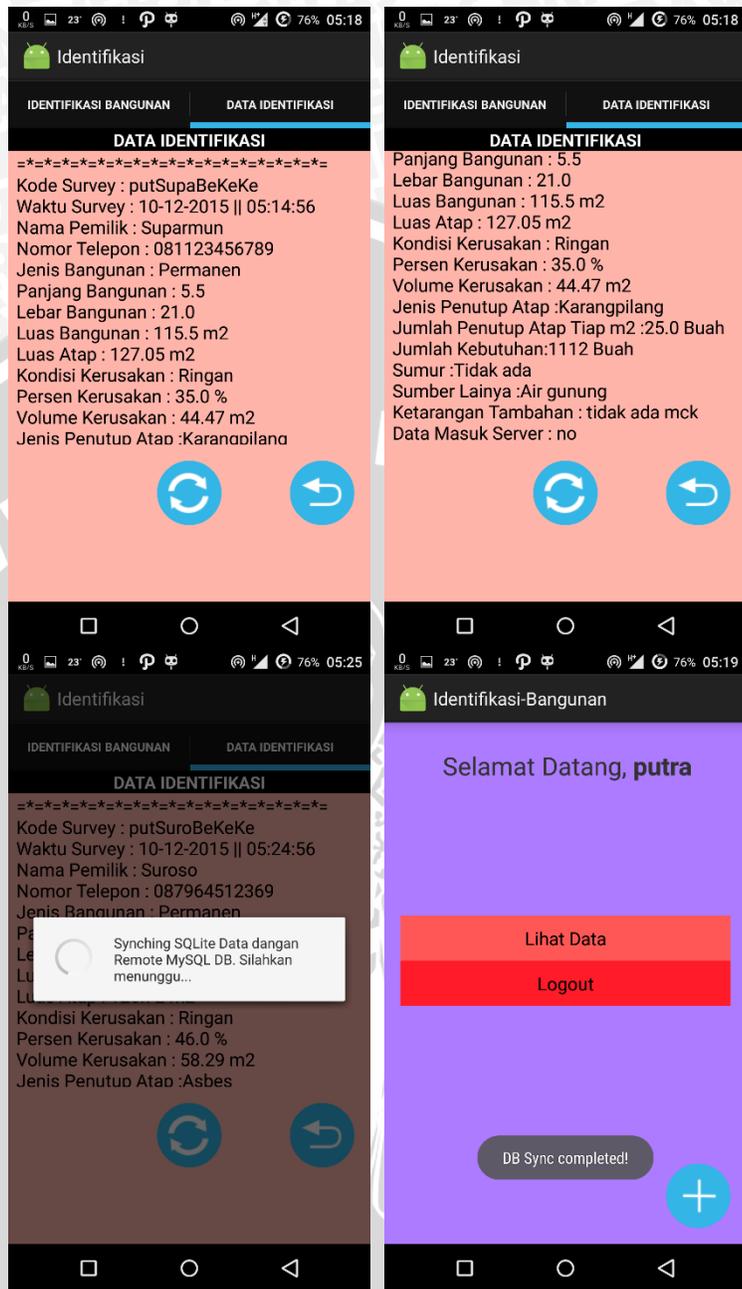
popup yang berisi tombol untuk melakukan perubahan data ataupun menghapus data.



Gambar 5.5. Hasil Implementasi Halaman *Edit Data*

f. Halaman *Sync Data*

Sync Data merupakan halaman yang akan menampilkan data yang akan di kirimkan ke *server* komputer. Yang mana data tersebut sudah tidak dapat lagi dirubah. Pada halaman ini juga terdapat tombol yang digunakan untuk mengirimkan data tersebut ke server komputer. Untuk mengirim data ke server pengguna harus menekan tombol *Sync Data* yang terdapat di halaman tersebut. Jika data tersebut berhasil dikirimkan ke server maka akan muncul pemberitahuan bahwa data tersebut telah terkirim dan masuk ke basis data pada server komputer.



Gambar 5.6. Hasil Implementasi Halaman Sync Data

g. Halaman View Data pada Server

View Data pada Server merupakan halaman yang akan menampilkan data hasil identifikasi kerusakan bangunan yang telah dimasukkan oleh pengguna ke dalam sistem aplikasi pada perangkat bergerak.

0 KB/S 23° 4G 72 06:01

www.afayuraputra.esy.es/Server/dataIdentifikasi/viewider

REFRESHI PENGARAH

Data Hasil Identifikasi Kekuasaan Bangunan

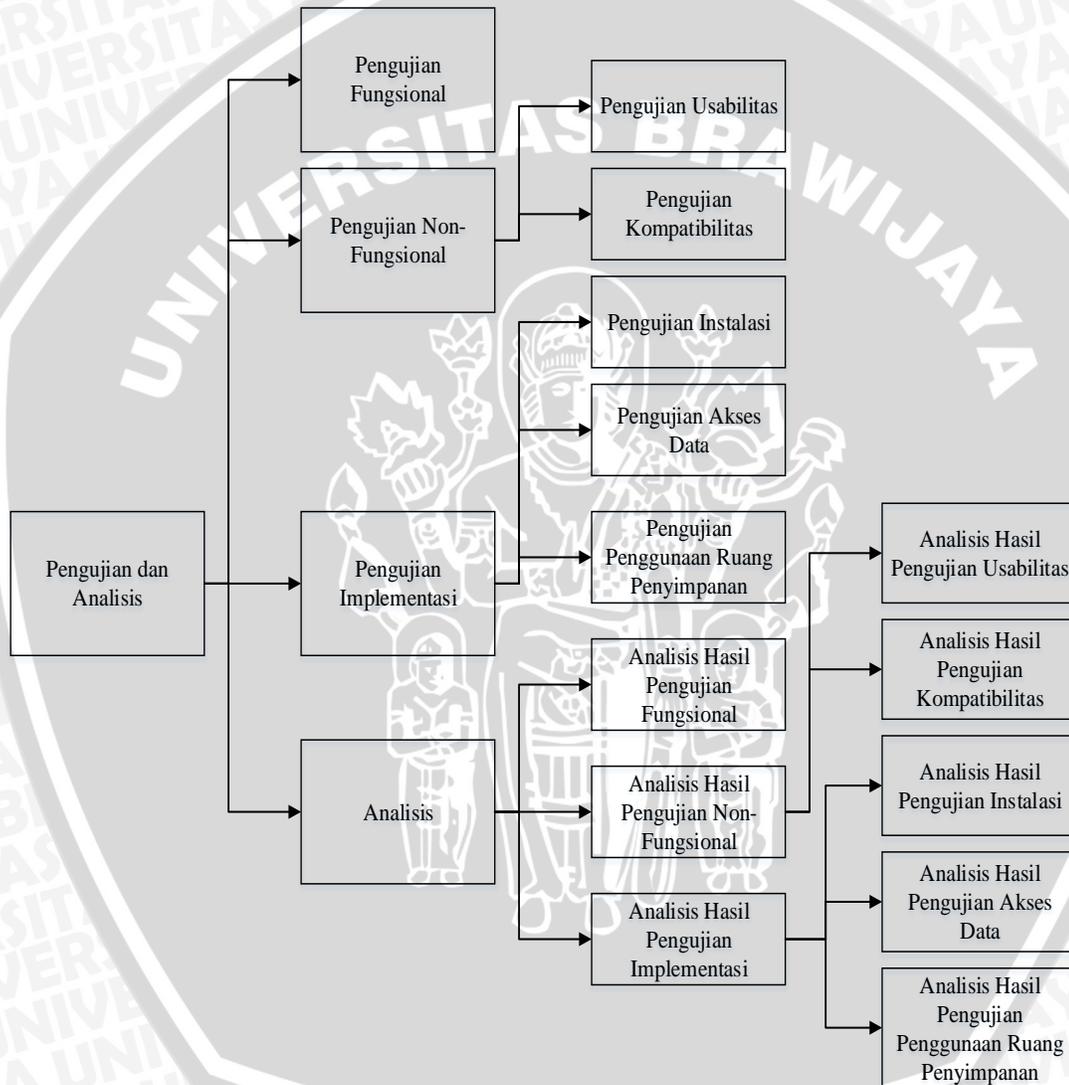
No	Waktu Survey	Nama Survejar	Kode Survey	Nama Pemilik	Telpom Pemilik	Nama Desa	Nama Kecamatan	Nama Wilayah	Jenis Bangunan	Panjang Bangunan	Lebar Bangunan	Luas Bangunan	Luas Atap Bangunan	Kondisi Kekuasaan	Persean Kekuasaan	Volume Kekuasaan	Jenis Pemenuh Atap	Jumlah Pemenuh Atap	Jumlah Kebutuhan	Sumur	Sumber Air Lain	Keterangan Tambahan	Foto Bangunan
1	10-12-2018 131348	Patra	patra	pujiBunFadu	085463281378348	Pabon	Sumenep	Sumenep	Permanen	5.5 m	25 m	137.5 m ²	151.3 m ²	Ringan	50%	75.6 m ³	Press / Loker	25	1891 Bush	ada	pdam	tidak ada	
2	13-12-2018 152728	Ala Yura Patra	AlaMarBeKeko	Marjon	085440509685	Besowo	Kepung	Kediri	Permanen	5.5 m	14 m	77 m ²	84.7 m ²	Ringan	30%	25.4 m ³	Press / Loker	25	636 Bush	tidak ada	air punung	tidak ada	
3	13-12-2018 153751	Ala Yura Patra	AlaPanBeKeko	Paimo	081123496789	Besowo	Kepung	Kediri	Permanen	6.4 m	12 m	76.8 m ²	84.5 m ²	Ringan	35%	29.6 m ³	Aubes	0.6	19 Bush	ada	tidak ada	tidak ada	

Gambar 5.7. Hasil Implementasi Halaman View Data pada Server



BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini, proses pengujian dan analisis hasil implementasi dari bab sebelumnya dilakukan. Pada Gambar 6.1 akan dijelaskan bagaimana struktur dari proses pengujian dan analisis dari aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana berbasis perangkat bergerak yang digambarkan dalam bentuk suatu pohon diagram.



Gambar 6.1. Pohon Diagram Pengujian dan Analisis

Proses pengujian terdiri dari beberapa jenis pengujian, yaitu: pengujian fungsional, pengujian non-fungsional, dan pengujian implementasi. Sedangkan analisis terdiri dari analisis pengujian fungsional, analisis pengujian non-fungsional, dan analisis pengujian implementasi.

6.1. Pengujian Fungsional

Pengujian fungsional digunakan untuk mengetahui apakah sistem yang dibangun sudah memiliki fungsi yang benar dan sesuai dengan apa yang dibutuhkan. Acuan dari pengujian ini adalah *Use Case* yang terdapat pada daftar kebutuhan fungsional yang telah di tuliskan sebelumnya. Pada pengujian ini akan digunakan teknik pengujian *black box*. Yang mana pada setiap kebutuhan fungsional akan dilakukan proses pengujian dengan kasus uji yang berbeda-beda. Dalam hal ini akan digambarkan pada Tabel 6.1 sampai Tabel 6.7.

Tabel 6.1. Kasus Uji Fungsional *Login*

Nomor Kasus Uji	UFU-01
Nama Kasus Uji	<i>Login</i>
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Untuk memastikan bahwa aplikasi perangkat bergerak dapat memenuhi kebutuhan fungsional dalam menyediakan form <i>login</i> , sehingga pengguna dapat memasukkan data alamat <i>email</i> dan <i>password</i> , untuk dapat masuk ke dalam sistem aplikasi perangkat bergerak.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka aplikasi. 2. Masuk ke halaman utama. 3. Tekan tombol <i>Login</i> 4. Masuk ke halaman <i>login</i> 5. Memasukkan alamat <i>email</i> dan <i>password</i> 6. Tekan tombol <i>login</i>
Hasil yang Diharapkan	Pengguna dapat membuka form <i>login</i> dan mengisi data alamat <i>email</i> dan <i>password</i> di form tersebut, sehingga dapat masuk ke dalam sistem aplikasi perangkat bergerak.

Tabel 6.2. Kasus Uji Fungsional *Register*

Nomor Kasus Uji	UFU-02
Nama Kasus Uji	<i>Register</i>
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Untuk memastikan bahwa aplikasi perangkat bergerak dapat memenuhi kebutuhan fungsional dalam menyediakan form <i>Register</i> , sehingga pengguna dapat memasukkan data identitas, untuk dapat terdaftar ke dalam sistem aplikasi perangkat

	bergerak.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka aplikasi. 2. Masuk ke halaman utama. 3. Tekan tombol <i>Login</i> 4. Masuk ke halaman <i>login</i> 5. Tekan tombol <i>register</i> 6. Masuk ke halaman registrasi 7. Memasukkan data identitas diri 8. Tekan tombol daftar
Hasil yang Diharapkan	Pengguna dapat membuka form <i>register</i> dan mengisi data identitas diri di form tersebut, sehingga dapat terdaftar ke dalam sistem aplikasi perangkat bergerak.

Tabel 6.3. Kasus Uji Fungsional Tambah Data Identifikasi

Nomor Kasus Uji	UFU-03
Nama Kasus Uji	Tambah Data Identifikasi
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Untuk memastikan bahwa aplikasi perangkat bergerak dapat memenuhi kebutuhan fungsional dalam menyediakan form <i>input</i> data identifikasi, sehingga pengguna dapat memasukkan data identifikasi meliputi data identitas bangunan, data kerusakan bangunan, serta data keterangan tambahan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka aplikasi. 2. Masuk ke halaman utama. 3. Tekan tombol <i>Login</i> 4. Masuk ke halaman <i>login</i> 5. Memasukkan alamat <i>email</i> dan <i>password</i> 6. Tekan tombol <i>login</i> 7. Masuk ke halaman identifikasi 8. Tekan tombol “+” untuk menambahkan data identifikasi. 9. Masuk ke halaman identifikasi kerusakan bangunan 10. Memasukkan data identifikasi kerusakan 11. Tekan tombol <i>submit</i>.
Hasil yang Diharapkan	Pengguna dapat membuka form <i>input</i> dan mengisi data identifikasi kerusakan bangunan di form tersebut, kemudian menyimpan data tersebut.

Tabel 6.4. Kasus Uji Fungsional Hapus Data Identifikasi

Nomor Kasus Uji	UFU-04
Nama Kasus Uji	Hapus Data Identifikasi
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Untuk memastikan bahwa aplikasi perangkat bergerak dapat memenuhi kebutuhan fungsional dalam menyediakan menu hapus data identifikasi , sehingga pengguna dapat menghapus data identifikasi yang telah dimasukkan ke dalam sistem aplikasi perangkat bergerak.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka aplikasi. 2. Masuk ke halaman utama. 3. Tekan tombol <i>Login</i> 4. Masuk ke halaman <i>login</i> 5. Memasukkan alamat <i>email</i> dan <i>password</i> 6. Tekan tombol <i>login</i> 7. Masuk ke halaman identifikasi 8. Pilih menu lihat data 9. Masuk ke halaman lihat data 10. Pilih data yang akan dihapus dengan menekan data tersebut hingga muncul menu hapus data 11. Tekan tombol hapus data
Hasil yang Diharapkan	Pengguna dapat menghapus data identifikasi yang sebelumnya telah dimasukkan ke dalam sistem aplikasi perangkat bergerak.

Tabel 6.5. Kasus Uji Fungsional Edit Data Identifikasi

Nomor Kasus Uji	UFU-05
Nama Kasus Uji	Edit Data Identifikasi
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Untuk memastikan bahwa aplikasi perangkat bergerak dapat memenuhi kebutuhan fungsional dalam menyediakan menu edit data identifikasi , sehingga pengguna dapat melakukan perubahan data identifikasi yang telah dimasukkan ke dalam sistem aplikasi perangkat bergerak.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka aplikasi. 2. Masuk ke halaman utama. 3. Tekan tombol <i>Login</i> 4. Masuk ke halaman <i>login</i> 5. Memasukkan alamat <i>email</i> dan <i>password</i>

	<ol style="list-style-type: none"> 6. Tekan tombol <i>login</i> 7. Masuk ke halaman identifikasi 8. Pilih menu lihat data 9. Masuk ke halaman lihat data 10. Pilih data yang akan di edit dengan menekan data tersebut hingga muncul menu edit data 11. Tekan tombol edit data 12. Masuk ke halaman edit data 13. Tekan tombol simpan
Hasil yang Diharapkan	Pengguna dapat melakukan perubahan data identifikasi yang sebelumnya telah dimasukkan ke dalam sistem aplikasi perangkat bergerak.

Tabel 6.6. Kasus Uji Fungsional Lihat Data Hasil Identifikasi

Nomor Kasus Uji	UFU-06
Nama Kasus Uji	Lihat Data Hasil Identifikasi
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Untuk memastikan bahwa aplikasi perangkat bergerak dapat memenuhi kebutuhan fungsional dalam menyediakan halaman lihat data hasil identifikasi, sehingga pengguna dapat melihat data hasil identifikasi yang telah dimasukkan ke dalam sistem aplikasi perangkat bergerak sebelumnya
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka aplikasi. 2. Masuk ke halaman utama. 3. Tekan tombol <i>Login</i> 4. Masuk ke halaman <i>login</i> 5. Memasukkan alamat <i>email</i> dan <i>password</i> 6. Tekan tombol <i>login</i> 7. Masuk ke halaman identifikasi 8. Pilih menu lihat data
Hasil yang Diharapkan	Pengguna dapat melihat data hasil identifikasi yang sebelumnya telah dimasukkan ke dalam sistem aplikasi perangkat bergerak.

Tabel 6.7. Kasus Uji Fungsional Kirim Data Hasil Identifikasi ke Server

Nomor Kasus Uji	UFU-07
Nama Kasus Uji	Kirim Data Hasil Identifikasi ke Server
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Untuk memastikan bahwa aplikasi perangkat bergerak dapat memenuhi kebutuhan fungsional dalam

	menyediakan menu untuk mengirimkan data hasil identifikasi ke server komputer.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka aplikasi. 2. Masuk ke halaman utama. 3. Tekan tombol <i>Login</i> 4. Masuk ke halaman <i>login</i> 5. Memasukkan alamat <i>email</i> dan <i>password</i> 6. Tekan tombol <i>login</i> 7. Masuk ke halaman identifikasi 8. Tekan tombol “+” 9. Masuk ke halaman identifikasi kerusakan bangunan 10. Pilih <i>tab</i> data identifikasi atau geser halamn ke kiri 11. Tekan tombol <i>sync</i>.
Hasil yang Diharapkan	Pengguna dapat mengirim data hasil identifikasi yang terdapat dalam basis data perangkat bergerak menuju ke server pada komputer.

6.2. Pengujian Non-Fungsional

Pengujian non-fungsional dilakukan dengan pengujian usabilitas dan kompatibilitas seperti yang telah didefinisikan pada perancangan sebelumnya.

6.2.1. Pengujian Usabilitas

Pengujian usabilitas dilakukan dengan menggunakan teknik kuisisioner, yang mana didalam kuisisioner tersebut terdapat 15 pertanyaan. Kuisisioner tersebut diberikan kepada beberapa orang responden. Responden – responden tersebut merupakan para *surveyor* yang dahulu pernah menggunakan sistem manual untuk melakukan identifikasi kerusakan bangunan.

Untuk mendapatkan umpan balik dari para responden, maka akan digunakan metode skala likert pada kuisisioner yang diberikan kepada responden tersebut. Skala Likert dirancang untuk meyakinkan para responden dalam menjawab berbagai tingkatan yang terdapat pada setiap pertanyaan ataupun pernyataan yang terdapat di dalam kuisisioner.

Dalam kuisisioner tersebut responden diminta untuk memberikan penilaian dengan cara menandai dengan tanda centang pada kolom – kolom yang berisi nilai angka 1 sampai dengan 5 pada setiap pertanyaan ataupun pernyataan yang diberikan. Dimana tiap nilai angka tersebut memiliki keterangan sebagai berikut.

- 1 = Sangat Tidak Setuju
- 2 = Tidak Setuju
- 3 = Netral

- 4 = Setuju
- 5 = Sangat Setuju

Kuesioner yang diberikan pada responden dibuat berdasarkan 5 komponen dari kualitas usability, yaitu *learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction* (Rianto, 2014). Yang mana susunan pertanyaan dari setiap kualitas tersebut akan dijelaskan sebagai berikut :

1. *Learnability*

Pada komponen ini, pertanyaan yang diberikan bertujuan untuk mengetahui tingkat kemudahan pengguna dalam menyelesaikan tugas – tugas dasar ketika pertamakali menggunakan aplikasi.

1. Apakah teks yang ada pada aplikasi jelas dan mudah dipahami?
2. Apakah menurut anda penamaan nama menu sudah tepat ?
3. Apakah informasi yang ditampilkan pada setiap halaman telah memungkinkan pengguna untuk dapat mengambil sebuah keputusan ?
4. Apakah judul pada tiap halaman cukup jelas dan informatif ?

2. *Efficiency*

Pada komponen ini, pertanyaan yang diberikan bertujuan untuk mengetahui tingkat kecepatan dalam menyelesaikan tugas-tugas yang ada saat pengguna pertamakali menggunakan aplikasi.

1. Apakah fitur-fitur yang disediakan dapat dengan mudah anda akses ?
2. Apakah penambahan data identifikasi pada halaman data identifikasi dapat anda lakukan dengan cepat ?
3. Apakah pada halaman lihat data dapat menampilkan data hasil identifikasi sesuai dengan ekspektasi anda ?

3. *Memorability*

Pada komponen ini, pertanyaan yang diberikan bertujuan untuk mengetahui seberapa mudah fitur yang terdapat di dalam aplikasi untuk diingat pengguna pada saat pengguna tersebut menggunakan aplikasi.

1. Apakah aplikasi mudah anda gunakan ?
2. Apakah penamaan nama menu dan judul halaman sesuai dengan ekspektasi yang anda inginkan ?
3. Apakah struktur menu dan halaman dapat dengan mudah anda ingat ?

4. *Errors*

Pada komponen ini, pertanyaan yang diberikan bertujuan untuk mengetahui seberapa mudah pengguna dapat mengatasi kesalahan apabila terjadi kesalahan yang dilakukan pengguna tersebut pada saat menggunakan aplikasi.

1. Apakah pesan kesalahan menginformasikan penyebab kesalahan ?
2. Apakah sistem mencegah anda ketika melakukan sebuah kesalahan ?

5. *Satisfaction*

Pada komponen ini, pertanyaan yang diberikan bertujuan untuk mengetahui tingkat kepuasan pengguna pada saat menggunakan aplikasi.

1. Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan ?
2. Apakah pengalaman dalam penggunaan aplikasi sudah memuaskan anda ?

6.2.2. Pengujian Kompatibilitas

Pengujian kompatibilitas digunakan untuk mengetahui kompatibilitas dari antarmuka sistem sesuai dengan yang dirancang yaitu untuk sistem operasi Android 4.4 keatas.

1. Pengujian Kompatibilitas Android versi 4.4

Pengujian kompatibilitas pada sistem operasi Android versi 4.4 dilakukan untuk mengetahui validitas dari kinerja fitur – fitur yang disediakan oleh sistem terkait dengan antarmuka sistem. Tabel 6.8 menjelaskan prosedur uji pengujian kompatibilitas pada sistem operasi Android versi 4.4

Tabel 6.8. Kasus Uji Kompatibilitas Sistem Operasi Android 4.4

Nama Kasus Uji	Pengujian Kompatibilitas Android 4.4
Objek Uji	Kebutuhan Non-Fungsional
Tujuan Pengujian	Untuk mengetahui validitas dari kinerja fitur – fitur yang disediakan oleh sistem terkait dengan antarmuka sistem.
Prosedur Pengujian	Membuka setiap halaman sesuai dengan spesifikasi kebutuhan sistem
Hasil yang Diharapkan	Sistem dapat menampilkan fitur – fitur dan antarmuka sesuai dengan implementasi pada antarmuka sistem

2. Pengujian Kompatibilitas Android versi 5.1.1

Pengujian kompatibilitas pada sistem operasi Android versi 5.1.1 dilakukan untuk mengetahui validitas dari kinerja fitur – fitur yang disediakan oleh sistem terkait dengan antarmuka sistem. Tabel 6.9 menjelaskan prosedur uji pengujian kompatibilitas pada sistem operasi Android versi 5.1.1

Tabel 6.9. Kasus Uji Kompatibilitas Sistem Operasi Android 5.1.1

Nama Kasus Uji	Pengujian Kompatibilitas Android 5.1.1
Objek Uji	Kebutuhan Non-Fungsional
Tujuan Pengujian	Untuk mengetahui validitas dari kinerja fitur – fitur yang disediakan oleh sistem terkait dengan antarmuka sistem.
Prosedur Pengujian	Membuka setiap halaman sesuai dengan spesifikasi kebutuhan sistem
Hasil yang Diharapkan	Sistem dapat menampilkan fitur – fitur dan antarmuka sesuai dengan implementasi pada antarmuka sistem

6.3. Pengujian Implementasi

Pada pengujian implementasi akan dilakukan 3 macam pengujian yaitu pengujian instalasi yang akan menguji apakah berkas aplikasi bisa dipasang pada perangkat bergerak, pengujian akses data menguji keluaran dari beberapa kondisi yang berlainan saat melakukan akses data dan pengujian penggunaan ruang penyimpanan, yang akan menguji dan menganalisa ruang penyimpanan yang dipakai dan dibutuhkan oleh aplikasi.

6.3.1. Pengujian Instalasi

Untuk melakukan pengujian instalasi, dibutuhkan berkas dari aplikasi perangkat bergerak untuk sistem operasi android yaitu berkas .apk. Pengujian instalasi ini nantinya akan memastikan bahwa berkas tersebut dapat dipasang dan berjalan di sistem operasi android. Tabel 6.10 menjelaskan hasil *build package* pada sistem operasi android.

Tabel 6.10. Hasil *Build Package* Pada Sistem Operasi Android.

Platform	Ukuran (MB)	Ekstensi
Android	1.81	.apk

Pada Tabel 6.11 akan dijelaskan mengenai prosedur uji dari pengujian instalasi pada sistem operasi android.

Tabel 6.11. Kasus Uji Instalasi pada Sistem Operasi Android

Nama Kasus Uji	Instalasi pada Sistem Operasi Android
Objek Uji	Implementasi
Tujuan Pengujian	Untuk memastikan bahwa aplikasi dapat dipasang dan berjalan pada sistem operasi android menggunakan

	berkas .apk
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Salin berkas .apk ke perangkat bergerak dengan sistem operasi android 2. Buka <i>file manager</i> 3. Cari dan pilih berkas .apk 4. Install berkas .apk tersebut 5. Buka aplikasi dengan memilih ikon aplikasi pada <i>app drawer</i> di perangkat bergerak
Hasil yang Diharapkan	Ikon aplikasi terlihat di <i>app drawer</i> pada perangkat bergerak dan aplikasi dapat berjalan sebagaimana mestinya.

6.3.2. Pengujian Akses Data

Pada pengujian akses data, akan dilakukan dengan cara melakukan percobaan akses data ke server yang terdapat di komputer (*web server*)

Ada beberapa kondisi dalam melakukan pengujian akses data, yaitu didalam kondisi terkoneksi *WiFi*, kondisi terkoneksi jaringan internet seluler, dan tanpa adanya koneksi internet. Tabel 6.12 sampai Tabel 6.14 akan menampilkan kasus uji yang berisi prosedur untuk menguji akses data dengan internet *WiFi*, internet seluler, dan tanpa koneksi internet.

Tabel 6.12. Kasus Uji Akses Data dengan Internet *WiFi*

Nama Kasus Uji	Akses Data dengan Internet <i>WiFi</i>
Objek Uji	Implementasi
Tujuan Pengujian	Untuk memastikan bahwa akses data dapat dilakukan dengan menggunakan koneksi internet <i>WiFi</i> dan mendapatkan informasi kecepatan akses data rata – rata dengan koneksi internet <i>WiFi</i>
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka menu pengaturan 2. Aktifkan pengaturan <i>WiFi</i> 3. Pilih <i>access point</i> 4. Buka aplikasi 5. Buka aplikasi dengan memilih ikon aplikasi pada <i>app drawer</i> di perangkat bergerak 6. Masuk ke halaman utama. 7. Tekan tombol <i>Login</i> 8. Masuk ke halaman <i>login</i> 9. Memasukkan alamat <i>email</i> dan <i>password</i> 10. Tekan tombol <i>login</i> 11. Masuk ke halaman identifikasi 12. Tekan tombol “+” untuk menambahkan data

	<p>identifikasi.</p> <p>13. Masuk ke halaman identifikasi kerusakan bangunan</p> <p>14. Memasukkan data identifikasi kerusakan</p> <p>15. Tekan tombol <i>submit</i>.</p>
Hasil yang Diharapkan	Akses data dengan menggunakan koneksi internet <i>WiFi</i> dapat dilakukan dengan baik

Tabel 6.13. Kasus Uji Akses Data dengan Internet Seluler

Nama Kasus Uji	Akses Data dengan Internet Seluler
Objek Uji	Implementasi
Tujuan Pengujian	Untuk memastikan bahwa akses data dapat dilakukan dengan menggunakan koneksi internet seluler.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka menu pengaturan 2. Aktifkan koneksi internet seluler 3. Buka aplikasi 4. Buka aplikasi dengan memilih ikon aplikasi pada <i>app drawer</i> di perangkat bergerak 5. Masuk ke halaman utama. 6. Tekan tombol <i>Login</i> 7. Masuk ke halaman <i>login</i> 8. Memasukkan alamat <i>email</i> dan <i>password</i> 9. Tekan tombol <i>login</i> 10. Masuk ke halaman identifikasi 11. Tekan tombol “+” untuk menambahkan data identifikasi. 12. Masuk ke halaman identifikasi kerusakan bangunan 13. Memasukkan data identifikasi kerusakan 14. Tekan tombol <i>submit</i>.
Hasil yang Diharapkan	Akses data dengan menggunakan koneksi internet Seluler dapat dilakukan dengan baik

Tabel 6.14. Kasus Uji Akses Data Tanpa Koneksi Internet

Nama Kasus Uji	Akses Data tanpa Koneksi Internet
Objek Uji	Implementasi
Tujuan Pengujian	Untuk memastikan bahwa akses data tidak dapat dilakukan dengan menggunakan koneksi internet seluler.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka menu pengaturan 2. Aktifkan koneksi internet seluler

	<ol style="list-style-type: none"> 3. Buka aplikasi 4. Buka aplikasi dengan memilih ikon aplikasi pada <i>app drawer</i> di perangkat bergerak 5. Masuk ke halaman utama. 6. Tekan tombol <i>Login</i> 7. Masuk ke halaman <i>login</i> 8. Memasukkan alamat <i>email</i> dan <i>password</i> 9. Tekan tombol <i>login</i>.
Hasil yang Diharapkan	Menampilkan pesan peringatan tidak ada koneksi internet.

6.3.3. Pengujian Penggunaan Ruang Penyimpanan

Pengujian ruang penyimpanan merupakan pengujian untuk menguji dan mendapatkan informasi banyaknya penggunaan kapasitas ruang penyimpanan yang dipakai oleh aplikasi. Ada dua macam penyimpanan yang akan diuji yaitu ruang penyimpanan yang digunakan aplikasi dan ruang penyimpanan yang digunakan oleh basis data. Tabel 6.15 dan Tabel 6.16 akan menampilkan prosedur uji yang digunakan untuk mendapatkan informasi ruang penyimpanan yang digunakan oleh aplikasi maupun basis data.

Tabel 6.15. Kasus Uji Penggunaan Ruang Penyimpanan Aplikasi

Nama Kasus Uji	Penggunaan Ruang Penyimpanan Aplikasi
Objek Uji	Implementasi
Tujuan Pengujian	Untuk mendapatkan informasi ruang penyimpanan yang digunakan aplikasi
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk ke halaman tambah data identifikasi 3. Tambahkan data identifikasi 4. Buka informasi aplikasi melalui menu pengaturan. 5. Kembali ke langkah 1 dan ulangi hingga 10 kali.

Tabel 6.16. Kasus Uji Penggunaan Ruang Penyimpanan Basis Data

Nama Kasus Uji	Penggunaan Ruang Penyimpanan Basis Data
Objek Uji	Implementasi
Tujuan Pengujian	Untuk mendapatkan informasi ruang penyimpanan yang digunakan basis data
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Buka aplikasi <i>android device monitor</i> 2. Pilih <i>package</i> dari aplikasi 3. Cari folder <i>databases</i>

	<ol style="list-style-type: none"> 4. Cari basis data dari aplikasi 5. Lihat ukuran berkas .db 6. Buka Aplikasi 7. Masuk ke halaman tambah data identifikasi 8. Tambahkan data identifikasi 9. Kembali ke langkah 6 dan berulang hingga 10 kali 10. Hapus Semua Data 11. Lihat ukuran berkas .db
--	--

Dari kasus uji yang ditampilkan pada Tabel 6.15, didapatkan hasil pengujian berupa penggunaan ruang penyimpanan internal pada aplikasi perangkat bergerak tiap terjadi penambahan 1 data, yang akan ditampilkan pada Tabel 6.7.

Tabel 6.17. Penggunaan Ruang Aplikasi Pada Media Penyimpanan Internal Perangkat Bergerak yang terpakai.

Jumlah Data	Ukuran Total Aplikasi (MB)
0	8.71
1	8.85
2	8.91
3	8.96
4	9.04
5	9.11
6	9.19
7	9.27
8	9.34
9	9.42
10	9.48

Selanjutnya dilakukan pengukuran volume basis data ketika aplikasi memiliki satu data, hingga beberapa data. Dari perosedur yang terdapat pada Tabel 6.16 didapatkan hasil pengujian berupa informasi ukuran basis data yang terpakai oleh aplikasi. Tabel 6.18 akan menampilkan informasi ukuran basis data yang terpakai.

Tabel 6.18. Ukuran Basis Data yang Terpakai

Jumlah Data	Ukuran Data (Bytes)
0	24576
1	28672
2	32768

3	36864
4	40960
5	45056
6	49152
7	53248
8	57344
9	61440
10	65536

Untuk tahap selanjutnya akan dilanjutkan proses pengujian untuk melihat ukuran basis data setelah data – data dihapus. Hal ini digunakan untuk memastikan bahwa data yang dihapus oleh pengguna tidak meninggalkan sisa dan memakai kapasitas ruang penyimpanan *internal*. Pada Tabel 6.19 akan menampilkan ukuran basis data setelah data terhapus seluruhnya.

Tabel 6.19. Ukuran Basis Data Setelah Data Terhapus

Jumlah Data	Ukuran Data (Bytes)
0	24576

6.4. Analisis

Proses analisis bertujuan untuk memperoleh hasil dan kesimpulan dari pengujian aplikasi yang telah dilakukan.

6.4.1. Analisis Pengujian Fungsional

Berdasarkan kasus uji fungsional pada masing-masing kebutuhan fungsional yang terdapat pada Tabel 6.1 sampai Tabel 6.7, maka diperoleh hasil pengujian fungsional seperti pada Tabel 6.20.

Tabel 6.20. Hasil Pengujian Fungsional

Nomor Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status
UFU-01	Pengguna dapat membuka form <i>login</i> dan mengisi data alamat <i>email</i> dan <i>password</i> di form tersebut, sehingga dapat masuk ke dalam sistem aplikasi perangkat bergerak.	Pengguna dapat membuka form <i>login</i> dan mengisi data alamat <i>email</i> dan <i>password</i> di form tersebut, sehingga dapat masuk ke dalam sistem aplikasi perangkat bergerak.	Valid
UFU-02	Pengguna dapat	Pengguna dapat	Valid

	membuka form <i>register</i> dan mengisi data identitas diri di form tersebut, sehingga dapat terdaftar ke dalam sistem aplikasi perangkat bergerak.	membuka form <i>register</i> dan mengisi data identitas diri di form tersebut, sehingga dapat terdaftar ke dalam sistem aplikasi perangkat bergerak.	
UFU-03	Pengguna dapat membuka form <i>input</i> dan mengisi data identifikasi kerusakan bangunan di form tersebut, kemudian menyimpan data tersebut.	Pengguna dapat membuka form <i>input</i> dan mengisi data identifikasi kerusakan bangunan di form tersebut, kemudian menyimpan data tersebut.	Valid
UFU-04	Pengguna dapat menghapus data identifikasi yang sebelumnya telah dimasukkan ke dalam sistem aplikasi perangkat bergerak.	Pengguna dapat menghapus data identifikasi yang sebelumnya telah dimasukkan ke dalam sistem aplikasi perangkat bergerak.	Valid
UFU-05	Pengguna dapat melakukan perubahan data identifikasi yang sebelumnya telah dimasukkan ke dalam sistem aplikasi perangkat bergerak.	Pengguna dapat melakukan perubahan data identifikasi yang sebelumnya telah dimasukkan ke dalam sistem aplikasi perangkat bergerak.	Valid
UFU-06	Pengguna dapat melihat data hasil identifikasi yang sebelumnya telah dimasukkan ke dalam sistem aplikasi perangkat bergerak.	Pengguna dapat melihat data hasil identifikasi yang sebelumnya telah dimasukkan ke dalam sistem aplikasi perangkat bergerak.	Valid
UFU-07	Pengguna dapat mengirim data hasil identifikasi yang terdapat dalam basis data perangkat bergerak menuju ke	Pengguna dapat mengirim data hasil identifikasi yang terdapat dalam basis data perangkat bergerak menuju ke	Valid

	server pada komputer.	server pada komputer.	
--	-----------------------	-----------------------	--

Pada Tabel 6.20 tersebut dapat terlihat bahwa keseluruhan hasil dari pengujian fungsional yang telah dilakukan menunjukkan status valid, hal tersebut membuktikan bahwa aplikasi yang dibuat ini telah sesuai dengan analisis kebutuhan fungsional yang telah didefinisikan pada Tabel 4.2.

6.4.2. Analisis Pengujian Non-Fungsional

Berdasarkan pengujian non-fungsional yang telah dilakukan sebelumnya, didapatkan analisis untuk pengujian usabilitas dan kompatibilitas.

6.4.2.1. Analisis Pengujian Usabilitas

Berdasarkan pengujian usabilitas melalui kuisisioner yang telah disebarakan kepada beberapa responden (detail jawaban terlampir pada lampiran), maka akan didapatkan hasil dari pengujian usabilitas yang dinyatakan pada Tabel 6.21.

Tabel 6.21. Hasil Pengujian Usabilitas

NO	Pertanyaan	1	2	3	4	5
1	Apakah teks yang ada pada aplikasi jelas dan mudah dipahami?	0	0	6	9	14
2	Apakah menurut anda penamaan nama menu sudah tepat ?	0	1	3	20	5
3	Apakah informasi yang ditampilkan pada setiap halaman telah memungkinkan pengguna untuk dapat mengambil sebuah keputusan ?	0	1	9	6	13
4	Apakah judul pada tiap halaman cukup jelas dan informatif ?	0	1	6	10	12
5	Apakah fitur-fitur yang disediakan dapat dengan mudah anda akses ?	0	2	4	14	9
6	Apakah penambahan data identifikasi pada halaman data identifikasi dapat anda lakukan dengan cepat ?	0	1	5	15	8
7	Apakah pada halaman lihat data dapat menampilkan data hasil identifikasi sesuai	0	1	6	14	8

	dengan ekspektasi anda ?					
8	Apakah aplikasi mudah anda gunakan ?	0	0	7	9	13
9	Apakah penamaan nama menu dan judul halaman sesuai dengan ekspektasi yang anda inginkan ?	0	1	7	12	9
10	Apakah struktur menu dan halaman dapat dengan mudah anda ingat ?	0	1	4	13	11
11	Apakah ikon-ikon yang terdapat pada aplikasi dapat mudah anda kenali?	0	2	6	11	10
12	Apakah pesan kesalahan menginformasikan penyebab kesalahan ?	0	0	7	14	8
13	Apakah sistem mencegah anda ketika melakukan sebuah kesalahan ?	0	1	4	16	8
14	Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan ?	0	0	8	12	9
15	Apakah pengalaman dalam penggunaan aplikasi sudah memuaskan anda ?	0	0	7	15	7

Proses analisis terhadap hasil pengujian usabilitas dilakukan dengan menggunakan skala likert. Persentase dari setiap pertanyaan akan ditunjukkan pada Tabel 6.22. Lalu hasil dari perhitungan indeks presentase dari setiap pertanyaan akan ditunjukkan pada Tabel 6.23. Dan hasil dari status pengujian usabilitas akan ditunjukkan pada Tabel 6.24. Untuk rumus perhitungan total skor akan ditunjukkan pada Persamaan 6.1, sedangkan rumus perhitungan indeks presentase ditunjukkan pada Persamaan 6.2.

Tabel 6.22. Presentase Skor Likert

Skor Likert	Interpretasi Skor Dengan Interval = 20	Pilihan
1	0% - 19,99%	Sangat Tidak Memuaskan
2	20% - 39,99%	Tidak Memuaskan
3	40% - 59,99%	Biasa
4	60% - 79,99%	Memuaskan
5	80% - 100%	Sangat Memuaskan

Tabel 6.23. Indeks Presentase

No	Pertanyaan	STS	TS	N	S	SS	Total Skor	Indeks (%)
<i>Learnability</i>								
1	Apakah teks yang ada pada aplikasi jelas dan mudah dipahami?	0	0	6	9	14	124	85.51
2	Apakah menurut anda penamaan nama menu sudah tepat ?	0	1	3	20	5	116	80
3	Apakah informasi yang ditampilkan pada setiap halaman telah memungkinkan pengguna untuk dapat mengambil sebuah keputusan ?	0	1	9	6	13	118	81.38
4	Apakah judul pada tiap halaman cukup jelas dan informatif ?	0	1	6	10	12	120	82.76
<i>Efficiency</i>								
5	Apakah fitur-fitur yang disediakan dapat dengan mudah anda akses ?	0	2	4	14	9	117	80.69
6	Apakah penambahan data identifikasi pada halaman data identifikasi dapat anda lakukan dengan cepat ?	0	1	5	15	8	117	80.69

7	Apakah pada halaman lihat data dapat menampilkan data hasil identifikasi sesuai dengan ekspektasi anda ?	0	1	6	14	8	116	80
<i>Memorability</i>								
8	Apakah aplikasi mudah anda gunakan ?	0	0	7	9	13	122	84.13
9	Apakah penamaan nama menu dan judul halaman sesuai dengan ekspetasi yang anda inginkan ?	0	1	7	12	9	116	80
10	Apakah struktur menu dan halaman dapat dengan mudah anda ingat ?	0	1	4	13	11	121	83.44
11	Apakah ikon-ikon yang terdapat pada aplikasi dapat mudah anda kenali?	0	2	6	11	10	116	80
<i>Errors</i>								
12	Apakah pesan kesalahan menginformasikan penyebab kesalahan ?	0	0	7	14	8	117	80.69
13	Apakah sistem mencegah anda ketika melakukan sebuah kesalahan ?	0	1	4	16	8	118	81.38
<i>Satisfaction</i>								

14	Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan ?	0	0	8	12	9	117	80.69
15	Apakah pengalaman dalam penggunaan aplikasi sudah memuaskan anda ?	0	0	7	15	7	116	80

Keterangan:

STS = Sangat Tidak Setuju N = Netral SS = Sangat Setuju
 TS = Tidak Setuju S = Setuju

$$\text{Total Skor} = (S_{STS} \times 1) + (S_{TS} \times 2) + (S_N \times 3) + (S_S \times 4) + (S_{SS} \times 5) \dots(6.1)$$

$$\text{Index (\%)} = (\text{Total Skor} / Y) \times 100 \dots\dots\dots(6.2)$$

Y = Skor Likert Tertinggi x Jumlah Responden

Tabel 6.24. Hasil Status Pengujian Usabilitas

Aspek Penilaian	Rata – Rata Presentase	Status
<i>Learnability</i>	82.41 %	Sangat Memuaskan
<i>Efficiency</i>	80.46 %	Sangat Memuaskan
<i>Memorability</i>	81.89 %	Sangat Memuaskan
<i>Error</i>	81.03 %	Sangat Memuaskan
<i>Satisfication</i>	80.34 %	Sangat Memuaskan
Rata - Rata	81.23 %	Sangat Memuaskan

Dari Hasil pengujian usabilitas yang telah dilakukan, menyatakan bahwa indeks persentase (%) yang didapatkan pada setiap aspek penilaian usabilitas, menunjukkan hasil rata – rata sebesar 81.23 % yaitu dengan status sangat memuaskan. Dengan adanya hasil tersebut, menunjukkan bahwa aplikasi telah sesuai dengan tujuan yang diinginkan.Salah satunya yaitu proses untuk mengetahui kebutuhan material akan lebih cepat dan efisien.

6.4.2.2. Analisis Hasil Pengujian Kompatibilitas

Proses analisis hasil pengujian kompatibilitas yang dilakukan pada aplikasi terhadap sistem operasi android dengan versi 4.4 dan versi 5.1.1

dilakukan untuk melihat kesesuaian antara hasil kinerja aplikasi dengan daftar kebutuhan. Hasil pengujian ditunjukkan dalam Tabel 6.25 dan 6.26

Tabel 6.25. Hasil Status Pengujian Kompatibilitas pada Android Versi 4.4

Nama Kasus Uji	Pengujian Kompatibilitas Android 4.4
Hasil yang Diharapkan	Sistem dapat menampilkan fitur – fitur dan antarmuka sesuai dengan implementasi pada antarmuka sistem
Hasil Yang didapatkan	Sistem dapat menampilkan fitur – fitur dan antarmuka sesuai dengan implementasi pada antarmuka sistem
Status Validitas	Valid

Tabel 6.26. Hasil Status Pengujian Kompatibilitas pada Android Versi 5.1.1

Nama Kasus Uji	Pengujian Kompatibilitas Android 5.1.1
Hasil yang Diharapkan	Sistem dapat menampilkan fitur – fitur dan antarmuka sesuai dengan implementasi pada antarmuka sistem
Hasil Yang didapatkan	Sistem dapat menampilkan fitur – fitur dan antarmuka sesuai dengan implementasi pada antarmuka sistem
Status Validitas	Valid

Pada Tabel 6.25 dan Tabel 6.26 dapat terlihat bahwa hasil pengujian kompatibilitas pada kedua tabel tersebut menunjukkan status valid, yaitu aplikasi perangkat bergerak identifikasi dan klasifikasi kerusakan rumah warga dampak bencana ini dapat dipasangkan dengan sesuai pada perangkat dengan sistem operasi android 4.4 dan android 5.1.1 dan dapat menampilkan fitur-fitur serta antarmuka sesuai dengan implementasi yang telah dilakukan, dari hal tersebut dapat disimpulkan bahwa aplikasi ini telah sesuai dengan parameter *compatibility* yang terdapat pada Tabel 4.10 di dalam analisis pengujian non-fungsional yaitu minimal aplikasi dapat dijalankan pada sistem operasi android dengan versi 4.4.

6.4.3. Analisis Hasil Pengujian Implementasi

Analisis pengujian implementasi menjelaskan analisis hasil dari pengujian instalasi pengujian akses data dan pengujian ruang penyimpanan.

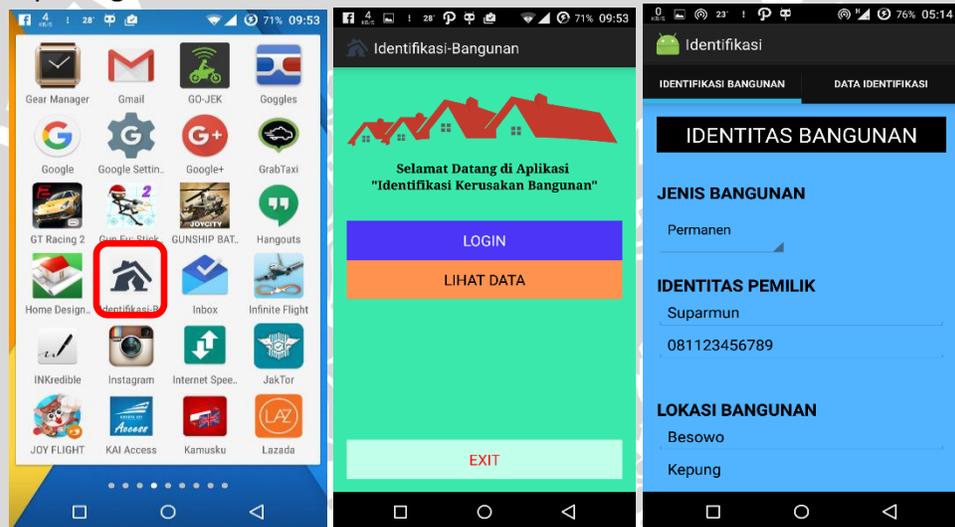
6.4.3.1. Analisis Pengujian Instalasi

Pada Tabel 6.27 akan menjelaskan hasil dari pengujian instalasi yang dilakukan. Dari kasus uji didapatkan hasil valid untuk pengujian instalasi.

Tabel 6.27. Hasil Pengujian Instalasi

Nama Kasus Uji	Instalasi pada Sistem Operasi Android
Hasil yang Diharapkan	Ikon aplikasi terlihat di <i>app drawer</i> pada perangkat bergerak dan aplikasi dapat berjalan sebagaimana mestinya.
Hasil Yang didapatkan	Ikon aplikasi terlihat di <i>app drawer</i> pada perangkat bergerak dan aplikasi dapat berjalan sebagaimana mestinya.
Status Validitas	Valid

Pada Gambar 6.2 akan menampilkan tangkapan layar hasil pengujian instalasi pada sistem operasi android, dimana ikon dengan kotak berwarna merah merupakan ikon dari aplikasi yang telah berhasil dipasang.



Gambar 6.2. Aplikasi Terpasang di Perangkat Bergerak Android

Pada Tabel 6.25 dapat terlihat bahwa hasil pengujian instalasi yang telah dilakukan menunjukkan status valid, yaitu aplikasi perangkat bergerak identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana ini dapat dipasang dengan lancar pada perangkat bergerak dengan sistem operasi *android*, sesuai dengan yang ditunjukkan pada Gambar 4.2. Hal tersebut membuktikan bahwa aplikasi perangkat bergerak ini telah sesuai dengan implementasi yaitu aplikasi ini dibuat untuk berjalan pada sistem operasi *android*.

6.4.3.2. Analisis Pengujian Akses Data

Pada Tabel 6.28 sampai Tabel 6.30 akan menjelaskan hasil dari pengujian akses data yang dilakukan. Kondisi yang digunakan adalah

kondisi terhubung dengan koneksi internet *WiFi*, terhubung dengan koneksi internet seluler, dan tidak terhubung dengan koneksi internet.

Tabel 6.28. Hasil Pengujian Akses Data dengan Internet *WiFi*

Nama Kasus Uji	Akses Data dengan Internet <i>WiFi</i>
Hasil yang Diharapkan	Akses data dengan menggunakan koneksi internet <i>WiFi</i> dapat dilakukan dengan baik
Hasil Yang didapatkan	Akses data dengan menggunakan koneksi internet <i>WiFi</i> dapat dilakukan dengan baik aplikasi dapat berjalan sebagaimana mestinya.
Status Validitas	Valid

Tabel 6.29. Hasil Pengujian Akses Data dengan Internet Seluler

Nama Kasus Uji	Akses Data dengan Internet Seluler
Hasil yang Diharapkan	Akses data dengan menggunakan koneksi internet Seluler dapat dilakukan dengan baik
Hasil Yang didapatkan	Akses data dengan menggunakan koneksi internet Seluler dapat dilakukan dengan baik
Status Validitas	Valid

Tabel 6.30. Hasil Pengujian Akses Data Tanpa Koneksi Internet

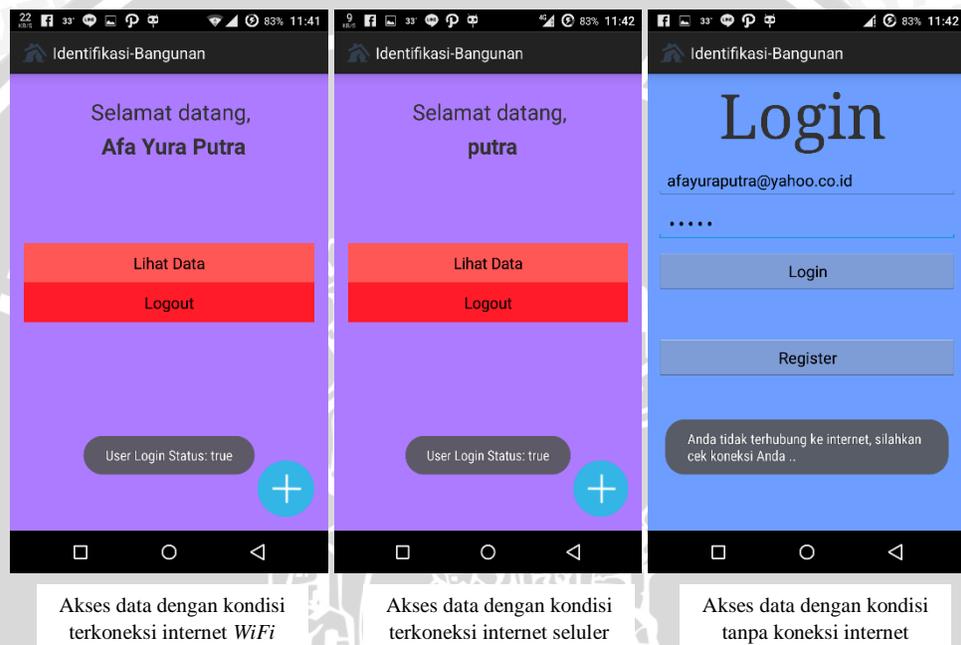
Nama Kasus Uji	Akses Data Tanpa Koneksi Internet
Hasil yang Diharapkan	Menampilkan pesan peringatan tidak ada koneksi internet.
Hasil Yang didapatkan	Menampilkan pesan peringatan tidak ada koneksi internet.
Status Validitas	Valid

Pada Gambar 6.3 akan menampilkan perbandingan hasil akses data pada halaman *login* ketika perangkat bergerak terhubung melalui koneksi internet *WiFi*, koneksi internet seluler, dan tanpa adanya koneksi internet.

Kondisi akses data ketika perangkat bergerak terhubung melalui koneksi internet baik melalui *WiFi* ataupun internet seluler, maka proses *login* akan berjalan sesuai dengan implementasi yang telah dilakukan, sedangkan jika perangkat bergerak yang digunakan tidak terhubung dengan koneksi internet, maka pada aplikasi akan muncul sebuah pemberitahuan bahwa perangkat bergerak yang digunakan sedang tidak terhubung ke jaringan internet.

Pada Tabel 6.28 sampai Tabel 6.30 terlihat bahwa hasil dari pengujian akses data menunjukkan status valid, yaitu pengguna dapat melakukan proses *login* ke dalam aplikasi jika perangkat bergerak yang

digunakan terhubung melalui jaringan internet, baik melalui jaringan internet *WiFi* ataupun jaringan internet selular, jika pengguna melakukan proses *login* namun perangkat bergerak yang digunakan tidak terhubung ke jaringan internet, maka akan muncul peringatan bahwa perangkat yang digunakan tersebut tidak sedang terhubung ke jaringan internet, seperti yang terlihat pada Gambar 6.3. Hal tersebut menunjukkan bahwa aplikasi ini telah sesuai dengan *point* yang terdapat pada batasan implementasi yaitu aplikasi membutuhkan koneksi internet untuk melakukan proses login dan mengirim data identifikasi ke server.



Akses data dengan kondisi terkoneksi internet *WiFi*

Akses data dengan kondisi terkoneksi internet seluler

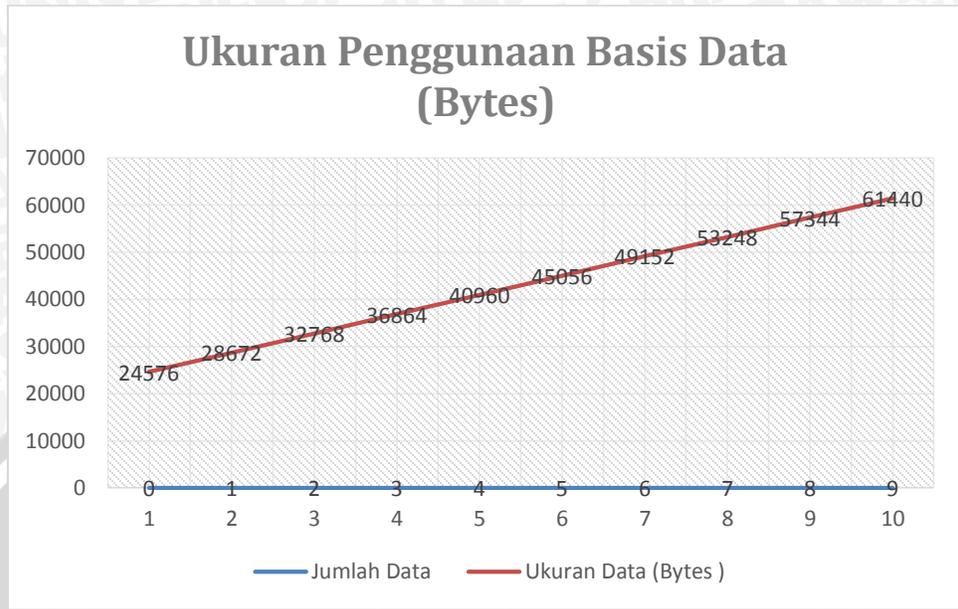
Akses data dengan kondisi tanpa koneksi internet

Gambar 6.3. Pengujian Akses Data dari Berbagai Kondisi

6.4.3.3. Analisis Hasil Pengujian Penggunaan Ruang Penyimpanan

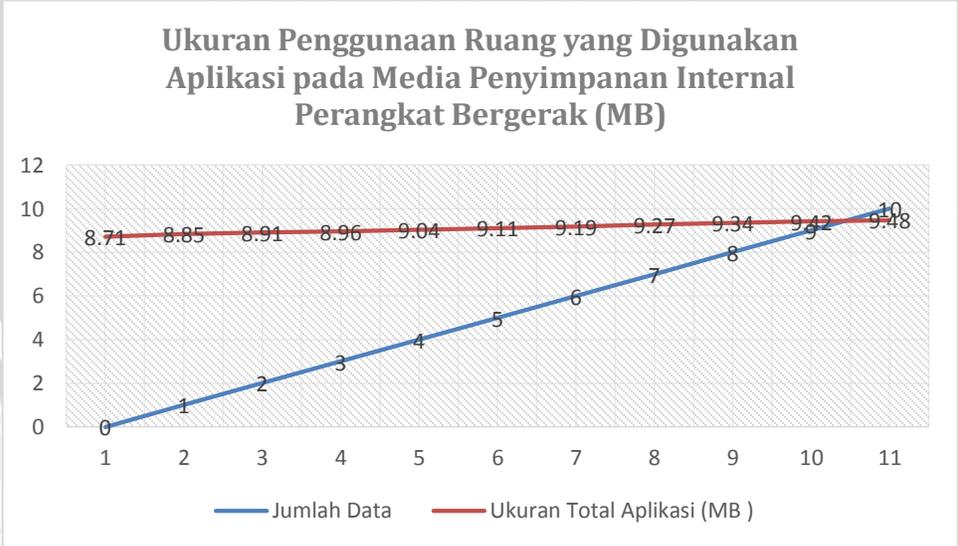
Pada pengujian penggunaan ruang penyimpanan, dilakukan pengukuran ruang penyimpanan basis data setiap terdapat penambahan data baru seperti yang terlihat pada Tabel 6.16.

Pada Gambar 6.4 akan menampilkan grafik ukuran basis data pada setiap terdapat penambahan data baru. Sedangkan pada Gambar 6.5



Gambar 6.4. Grafik Ukuran Penggunaan Basis Data

Dari grafik tersebut, terlihat bahwa pertambahan ukuran basis data bergerak secara linear. Ukuran basis data meningkat secara linear pada setiap penambahan data identifikasi, yaitu dengan penambahan ukuran penggunaan ruang penyimpanan basis data sebesar 4096 bytes setiap data yang ditambahkan ke dalam basis data.



Gambar 6.5. Grafik Ukuran Penggunaan Ruang yang Digunakan Aplikasi pada Media Penyimpanan Internal Perangkat Bergerak .

Dari grafik tersebut, terlihat bahwa pertambahan ukuran aplikasi bergerak secara linear. Ukuran aplikasi meningkat secara linear pada

setiap penambahan data identifikasi. Jika Semula sebelum terdapat data ukuran aplikasi sebesar 8.71 MB, namun setelah terjadi penambahan sebanyak 10 data, maka ukuran dari aplikasi akan meningkat menjadi sebesar 9.48 MB.

Semakin banyak data yang dimasukkan, maka akan semakin besar pula ruang media penyimpanan perangkat bergerak yang dibutuhkan, sebagai contoh jika kita ingin menambahkan 100 data ke dalam aplikasi, maka setidaknya ruang kosong pada media penyimpanan yang harus disediakan di dalam perangkat bergerak adalah sekitar 90.48 MB.

Maka dari itu, untuk menghindari kurang berfungsinya aplikasi pada saat melakukan penambahan data, maka pengguna harus menyesuaikan terlebih dahulu kira-kira berapa banyak data yang akan dimasukkan ke dalam sistem pada aplikasi dengan ruangan yang masih tersedia pada media penyimpanan internal dari perangkat bergerak yang dimilikinya, sehingga nantinya tidak ada masalah pada saat proses pemasukan data ke dalam sistem pada aplikasi perangkat bergerak yang digunakan.

Berdasarkan keseluruhan pengujian dan analisis yang telah dilakukan, dapat ditarik suatu kesimpulan bahwa aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana ini telah sesuai dengan perancangan dan implementasi yang telah dijabarkan sebelumnya.



BAB 7 KESIMPULAN DAN SARAN

7.1. Kesimpulan

Berdasarkan hasil analisis perancangan, implementasi, dan pengujian yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Perancangan aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana berbasis *mobile* telah sesuai dengan spesifikasi kebutuhan yang telah dianalisa.
2. Aplikasi identifikasi dan klasifikasi kerusakan rumah tinggal warga dampak bencana berbasis *mobile* berhasil menghasilkan rekomendasi data kebutuhan material berdasarkan data identifikasi kerusakan yang telah di masukkan ke dalam sistem.
3. Berdasarkan hasil pengujian fungsional dengan menggunakan metode *black box* menunjukkan bahwa semua fitur atau kebutuhan fungsional telah sesuai dengan yang dibutuhkan.
4. Berdasarkan aspek pengujian usabilitas yang meliputi aspek *learnability, efficiency, memorability, errors, dan satisfaction*, aplikasi memiliki rata – rata tingkat usabilitas sebesar 81.23% yaitu dengan status sangat memuaskan.
5. Berdasarkan hasil pengujian implementasi yang terdiri dari pengujian instalasi, akses data, dan penggunaan ruang penyimpanan, menunjukkan bahwa aplikasi dapat terpasang dan berjalan dengan baik serta mampu menangani akses data dari berbagai kondisi dan memastikan alur proses fungsional dalam aplikasi berjalan dengan normal dimana penggunaan kapasitas ruang untuk basis data dan aplikasi naik secara linear pada saat menambahkan data identifikasi kerusakan bangunan serta tidak ada jejak ruang yang tertinggal pada saat data identifikasi bangunan dihapus seluruhnya.

7.2. Saran

Berdasarkan saran yang telah dituliskan oleh para *surveyor* di dalam kuisisioner yang telah diberikan serta pengujian yang telah dilakukan, maka saran yang dapat diberikan untuk pengembangan aplikasi identifikasi kerusakan rumah warga dampak bencana selanjutnya, antara lain :

1. Pada fitur penambahan data identifikasi perlu dilakukan penambahan kriteria kerusakan bangunan, agar data kebutuhan material yang dihasilkan bisa lebih bermacam - macam lagi.
2. Tampilan di dalam aplikasi perlu ditingkatkan lagi agar isi di dalam aplikasi bisa lebih mudah untuk diakses.

3. Navigasi yang terdapat di dalam aplikasi perlu ditingkatkan lagi, agar pengguna bisa lebih mudah dalam menggunakan aplikasi.
4. Dilakukan penambahan dukungan untuk sistem operasi perangkat bergerak lainnya.



DAFTAR PUSTAKA

- BNPB (Badan Nasional Penanggulangan Bencana). 2011. Definisi BNPB. BNPB. Jakarta. (<http://www.bnpb.go.id>). Tanggal Akses 21 April 2015
- BNPB (Badan Nasional Penanggulangan Bencana). 2011. Potensi Ancaman Bencana. BNPB. Jakarta. (<http://www.bnpb.go.id/pengetahuan-bencana>). Tanggal Akses 26 Februari 2015
- BNPB (Badan Nasional Penanggulangan Bencana). 2011. Tugas dan Fungsi BNPB. BNPB. Jakarta. (<http://www.bnpb.go.id/profil/tugas-dan-fungsi>). Tanggal Akses 21 April 2015
- Fawzila, Rizki. 2012. Aplikasi MAP Kota Bandung *Offline* Berbasis Symbian. Bandung : Universitas Komputer Indonesia
- Hidayat, Hastyanti. 2008. Evaluasi Proyek Konstruksi Rumah Tahan Gempa Di Desa Cucukan, Prambanan, Klaten, Jawa Tengah. Semarang : Universitas Diponegoro
- Idham, Febrina Hasida. 2014. Identifikasi Kerusakan Komponen Bangunan Rumah Tinggal Akibat Gempa Dengan Tingkat Kerusakan Ringan Di Kabupaten Bener Meriah. Banda Aceh : Universitas Syiah Kuala
- Kartadie, Rikie dkk. 2013. Aplikasi Sistem Informasi Manajemen Penanggulangan Pasca Bencana. Yogyakarta : STMIK AMIKOM
- Laboratorium Manajemen Konstruksi. 2014. Tabel Perhitungan Volume Kerusakan dan Kebutuhan Material. Malang : Universitas Brawijaya.
- Mega, Dellyani. 2013. Definisi dan Fungsi Rumah Tinggal. Bandung : Universitas Pendidikan Indonesia.
- Nasaruddin dkk. 2011. Aplikasi Pelaporan Kejadian Bencana Secara *Online* Di Wilayah Aceh Berbasis SMS *Gateway*. Banda Aceh : Universitas Syiah Kuala.
- Pakpahan, Tigor dkk. 2013. Aplikasi *Mobile* Peta Rawan Bencana Kota Manado Berbasis Android. Manado : Universitas Sam Ratulangi.
- Pantita Teknis Konstruksi Bangunan. 2002. Kumpulan Analisa Biaya Konstruksi Bangunan Gedung Dan Perumahan. Bandung : Badan Standarisasi Nasional.
- Rianto, Dedi Rahardi. 2014. Pengujian *Usability* Sistem menggunakan *Use Questionnaire* Pada Aplikasi Android. Palembang
- Tim Engineer Fakultas Teknik. 2014. Identifikasi dan Klasifikasi Kerusakan Rumah Warga Dampak Erupsi Gunung Kelud. Malang : Universitas Brawijaya.
- Zaki, Ali. 2008. E-Life Style: Memanfaatkan Beragam Perangkat Teknologi Digital. Jakarta : Penerbit Salemba Infotek.

LAMPIRAN

Lampiran 1. Lembar Kuesioner Pengujian Usabilitas

Identifikasi Bangunan App

Nama :

No Identitas :

(KTP/NIM/NIP/NIK)

Email :

UMPMAN BALIK

Beri tanda centang (✓) pada masing-masing pertanyaan dengan jawaban yang menurut anda paling sesuai!

1 = Sangat Tidak Setuju 3 = Netral 5 = Sangat Setuju
2 = Tidak Setuju 4 = Setuju

NO	Pertanyaan	1	2	3	4	5
1	Apakah teks yang ada pada aplikasi jelas dan mudah dipahami?					
2	Apakah menurut anda penamaan nama menu sudah tepat ?					
3	Apakah informasi yang ditampilkan pada setiap halaman telah memungkinkan pengguna untuk dapat mengambil sebuah keputusan ?					
4	Apakah judul pada tiap halaman cukup jelas dan informatif ?					
5	Apakah fitur-fitur yang disediakan dapat dengan mudah anda akses ?					
6	Apakah penambahan data identifikasi pada halaman data identifikasi dapat anda lakukan dengan cepat ?					
7	Apakah pada halaman lihat data dapat menampilkan data hasil identifikasi sesuai dengan ekspektasi anda ?					
8	Apakah aplikasi mudah anda gunakan ?					

9	Apakah penamaan nama menu dan judul halaman sesuai dengan ekspektasi yang anda inginkan ?				
10	Apakah struktur menu dan halaman dapat dengan mudah anda ingat ?				
11	Apakah ikon-ikon yang terdapat pada aplikasi dapat mudah anda kenali?				
12	Apakah pesan kesalahan menginformasikan penyebab kesalahan ?				
13	Apakah sistem mencegah anda ketika melakukan sebuah kesalahan ?				
14	Apakah semua fitur yang ada telah mencakup informasi atau kebutuhan yang diharapkan ?				
15	Apakah pengalaman dalam penggunaan aplikasi sudah memuaskan anda ?				

Kritik :

.....
.....

Saran :

.....
.....

Malang,

(.....)

Lampiran 2. Hasil Umpan Balik Kuesioner

No	Nama	Pertanyaan														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Saifoe El Unas	3	2	4	2	3	4	3	3	2	2	2	4	4	3	3
2	S. Prayitno	4	5	4	5	5	4	4	5	4	4	4	5	5	5	5
3	Eva Arifi	5	4	5	5	4	4	5	5	4	5	5	4	3	5	5
4	Kartika Puspa Negara	5	4	5	4	5	5	4	5	4	4	5	5	3	5	5
5	Devi Nuralinah	5	5	4	5	5	4	4	5	5	4	5	5	3	4	5
6	Alwafi Pujiraharjo	5	4	4	5	5	4	5	5	4	5	5	5	4	5	5
7	Christin Remayanti N.	5	4	5	5	4	4	5	4	5	5	4	5	4	5	3
8	Rahayu Kusumaningrum	5	4	5	5	4	3	4	3	5	5	3	4	4	3	3
9	Noval	5	4	5	5	4	3	4	3	4	5	3	4	3	3	3
10	Roland Martin Simatupang	3	5	4	5	4	3	5	5	5	5	4	4	5	4	4
11	Indradi W	4	4	5	5	4	4	5	4	4	4	3	4	4	3	4
12	Desy Setyowulan	4	4	3	3	4	3	4	3	4	3	3	4	2	3	3
13	Ananda Insan Firdausy	4	4	4	5	4	5	4	5	5	4	5	3	4	4	4
14	Lilya Susanti	4	5	4	5	4	5	4	4	5	5	5	4	5	5	4
15	Eko Andi Suryo	5	5	5	5	4	5	5	5	5	4	5	4	4	5	4
16	Bhona	5	4	3	3	2	4	3	4	3	5	5	4	4	3	3
17	Ella Dwi W.	5	4	3	4	3	5	4	5	3	4	5	3	3	3	4

18	Irdwi Juni K.	3	4	3	3	2	3	3	3	3	3	4	2	4	4	3
19	Iqbal Habibi	4	4	4	4	4	4	2	4	4	4	2	3	4	3	4
20	Ming Narto Wijaya	5	4	4	4	5	4	4	5	5	4	4	4	4	5	4
21	Ketut Sugiarto	4	4	5	3	3	5	5	4	4	5	4	3	5	4	5
22	Awang Pradyka	5	4	2	3	3	2	4	4	5	4	4	3	4	4	4
23	Indra Waluyohadi	4	4	3	4	3	4	5	4	3	4	4	4	5	5	5
24	Rina Yunita	5	3	3	3	5	4	3	5	3	4	3	3	4	4	4
25	Dhino Teguh P.	3	4	3	4	4	4	4	3	4	3	3	3	4	4	4
26	Sukma Patrya	4	4	5	4	5	4	4	5	4	5	4	5	5	4	4
27	Agustina tri Yolanda	5	3	3	5	5	5	3	5	3	4	5	5	4	4	4
28	Sugeng Hendik	3	3	4	4	4	4	4	3	4	3	4	4	4	4	4
29	Alifha	3	4	5	4	4	5	3	4	3	5	4	3	4	4	4

Lampiran 3. Tabel Perhitungan Volume Kerusakan dan Kebutuhan Material

