

RANCANG BANGUN APLIKASI *MOBILE*
PANDUAN BUDIDAYA TANAMAN CABAI ORGANIK
SECARA MANDIRI

SKRIPSI

LABORATORIUM PERANGKAT BERGERAK

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh:

ADITIA SHELI CAHYA UTAMA

NIM. 105060807111010

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER

MALANG

2015

LEMBAR PERSETUJUAN
RANCANG BANGUN APLIKASI *MOBILE* PANDUAN BUDIDAYA
TANAMAN CABAI ORGANIK SECARA MANDIRI

SKRIPSI

LABORATORIUM PERANGKAT BERGERAK
Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

ADITIA SHELI CAHYA UTAMA

NIM. 105060807111010

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II

Herman Tolle, Dr. Eng, ST., MT.

NIP. 19740823 200012 1 001

Fajar Pradana, S.ST., M.Eng

NIK. 87112116110371

LEMBAR PENGESAHAN
RANCANG BANGUN APLIKASI *MOBILE* PANDUAN BUDIDAYA
TANAMAN CABAI ORGANIK SECARA MANDIRI

SKRIPSI

LABORATORIUM PERANGKAT BERGERAK

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun oleh :

ADITIA SHELI CAHYA UTAMA

NIM. 105060807111010

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 12 Januari 2015
dan dinyatakan memenuhi syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer

Penguji I

Tri Astoto Kurniawan, ST, MT, Ph.D.

NIP. 19710518 200312 1 001

Penguji II

Issa Arwani, S.Kom., M.Sc.

NIP. 19830922 201212 1 003

Penguji III

Wibisono Sukmo Wardhono, ST., MT

NIK. 820404 06 1 1 0091

Mengetahui
Ketua Program Studi Informatika / Ilmu Komputer

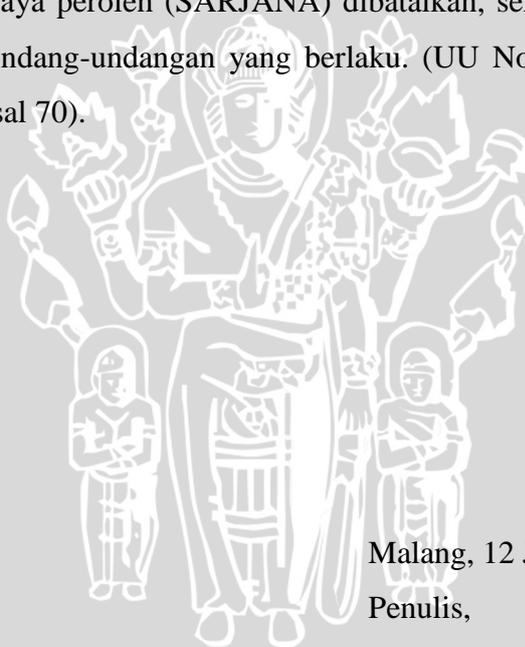
Drs. Marji, M.T.

NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

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

Apabila ternyata didalam naskah 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, 12 Januari 2015
Penulis,

Aditia Sheli Cahya Utama
NIM. 105060807111010

KATA PENGANTAR

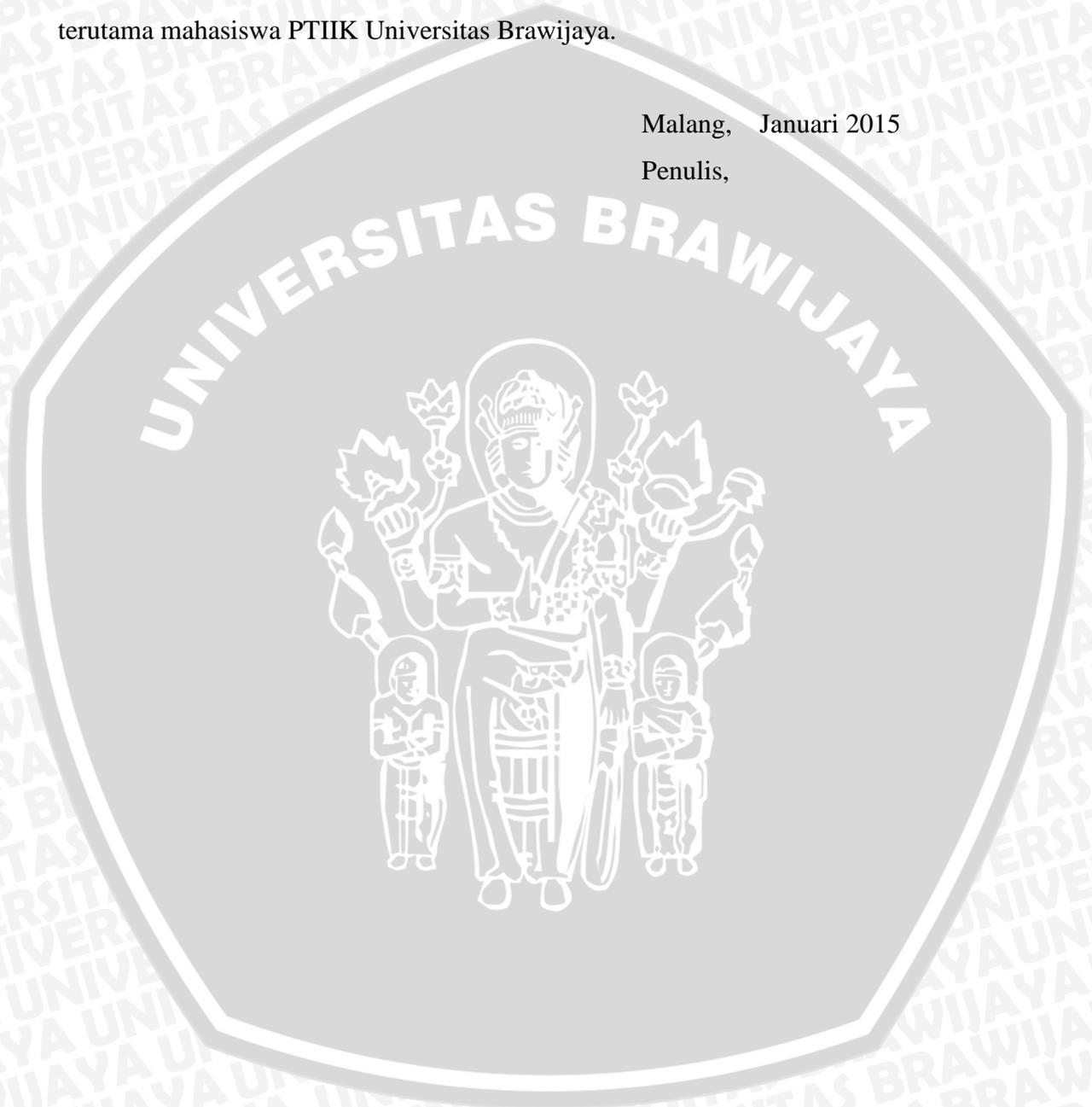
Puji dan syukur Penulis panjatkan kehadirat Allah SWT, karena hanya dengan rahmat dan karunia-Nya Penulis dapat menyelesaikan skripsi dengan judul **“RANCANG BANGUN APLIKASI *MOBILE* PANDUAN BUDIDAYA TANAMAN CABAI ORGANIK SECARA MANDIRI”** dengan baik. Melalui kesempatan ini, Penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama pengerjaan skripsi, diantaranya:

1. Kedua orang tua H. Senari, M.Pd. dan Yuliana, S.Pd. yang telah memberi motivasi, kasih sayang serta dukungan moril dan materil. Adik Adinda Sheli Dwita Sari serta Yeni Dwi Lestari yang telah memberikan semangat dari awal sampai akhir pengerjaan skripsi ini.
2. Bapak Herman Tolle, Dr. Eng, ST., MT. selaku dosen pembimbing I yang telah banyak memberikan ilmu dan saran selama penyusunan skripsi ini.
3. Bapak Fajar Pradana, S.ST., M.Eng. selaku dosen pembimbing II yang juga telah banyak memberikan ilmu dan saran selama penyusunan skripsi ini.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
5. Semua teman-teman PTIIK, khususnya Informatika/Ilmu Komputer angkatan 2010 terima kasih atas segala bantuan dan dukungannya selama ini.
6. Semua pihak yang tidak dapat penulis sebutkan satu-persatu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya skripsi ini.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT. Dengan segala kerendahan hati, penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari sempurna karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Akhirnya, semoga skripsi ini dapat bermanfaat dan berguna bagi pembaca terutama mahasiswa PTIIK Universitas Brawijaya.

Malang, Januari 2015

Penulis,



ABSTRAK

Aditia Sheli Cahya Utama. 2015. Rancang Bangun Aplikasi *Mobile* Panduan Budidaya Tanaman Cabai Organik Secara Mandiri. Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Herman Tolle, Dr. Eng, ST., MT. dan Fajar Pradana, S.ST., M.Eng.

Pola konsumsi organik telah menjadi tren gaya hidup sehat yang dipicu oleh kesadaran masyarakat akan penggunaan pestisida kimiawi pada pertanian konvensional berbahaya bagi kesehatan. Di Indonesia, komoditas hortikultura merupakan produk unggulan dan salah satu produk yang sangat dibutuhkan masyarakat Indonesia adalah cabai. Cabai merupakan salah satu bumbu yang banyak digunakan dalam berbagai masakan olahan nusantara. Sehingga keamanan produk ini perlu diperhatikan dari residu pestisida kimiawi, misalnya dengan beralih mengonsumsi produk cabai organik. Daripada membeli produk pertanian organik dengan harga mahal, lebih baik melakukan budidaya cabai organik secara mandiri. Berbagai referensi untuk melakukan budidaya secara mandiri masih dalam bentuk buku, sehingga memerlukan beberapa buku untuk mendapatkan informasi secara lengkap.

Penelitian ini bermaksud untuk memberikan informasi kebutuhan budidaya dalam sebuah perangkat *mobile* serta pemanfaatan teknologi *mobile* menjadi fitur-fitur yang memudahkan pengguna dalam melakukan proses budidaya secara mandiri. Aplikasi dibuat dengan *framework* phonegap menggunakan pendekatan pengembangan *hybrid* yang memungkinkan aplikasi native dibangun menggunakan bahasa pemrograman HTML, CSS dan javascript.

Berdasarkan hasil penelitian yang telah dilakukan, aplikasi dapat diterima sebagai solusi untuk melakukan budidaya cabai organik secara mandiri dengan memanfaatkan teknologi *mobile* untuk mendapatkan hasil optimal yang diukur dari kemudahan penggunaan dan pencapaian kegunaan dengan persentase sebesar 82,76%.

Kata kunci: Budidaya cabai organik, *hybrid*, *mobile*.

ABSTRACT

Aditia Sheli Cahya Utama. 2015. *Design and Implementation Mobile Application Guide Cultivation Organic Chili Independently*. Information Technology and Computer Science Program, Brawijaya University, Malang. Advisor: Herman Tolle, Dr. Eng, ST., MT. and Fajar Pradana, S.ST., M.Eng.

Organic consumption has become a trend of healthy lifestyle. That was triggered by the use of chemical pesticides in conventional agriculture can be harmful to health. Horticulture is an excellent product in Indonesia. Chili is a horticultural products that are needed by the Indonesian people. Because chili is one of the ingredients are mostly used in Indonesian cuisine. So that the safety of these products need to be considered from a chemical pesticide residues, for example by consuming organic chilli products. Instead of buying organic agricultural products with high prices better cultivate organic chili independently. Various references about self-cultivation is still a book, so that need some books to obtain complete information and becomes impractical.

This study intends to provide information cultivation in a mobile device application to make it more practical and utilizing mobile technology into features that allow users to perform self-cultivation process. The application has been made using PhoneGap framework hybrid development approach that enables native applications built using the programming language HTML, CSS and javascript.

Implementation of application was accepted as a solution in performing organic chili cultivation independently by utilizing mobile technology to get optimal results. That was measured from perceived ease of use and perceived usefulness with a percentage of 82.76%.

Keywords: *Hybrid, mobile, organic chili cultivation.*

DAFTAR ISI

LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN ORISINALITAS SKRIPSI.....	iv
KATA PENGANTAR.....	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	5
BAB II KAJIAN PUSTAKA DAN DASAR TEORI.....	6
2.1 Cabai.....	6
2.1.1 Cabai Rawit.....	6
2.1.2 Cabai Merah.....	7
2.2 Pertanian Organik.....	7
2.3 Teknik Budidaya Cabai Secara Organik	7
2.4 Pupuk Organik.....	8
2.5 Pestisida Alami.....	9
2.6 Android.....	10
2.7 <i>HyperText Markup Language</i> (HTML).....	10
2.8 <i>Cascading Style Sheet</i> (CSS).....	11
2.9 Javascript	12
2.10 JQuery	12
2.11 JQuery Mobile	13
2.12 JSON	13



2.13	SQLite	13
2.14	PhoneGap	14
2.15	Aptana Studio	16
2.16	XAMPP	16
2.17	Rekayasa Perangkat Lunak.....	16
2.17.1	Analisis Kebutuhan (<i>Requirement Analysis</i>)	17
2.17.2	Perancangan (Design)	18
2.17.3	Implementasi.....	18
2.17.4	Pengujian	18
BAB III METODE PENELITIAN		20
3.1	Studi Literatur.....	20
3.2	Analisis Kebutuhan	21
3.3	Perancangan Sistem.....	21
3.4	Implementasi	22
3.5	Pengujian	23
3.6	Pengambilan Kesimpulan dan Saran	23
BAB IV ANALISIS DAN PERANCANGAN		24
4.1	Analisis Kebutuhan	25
4.1.1	Analisis Fitur Pilihan	25
4.1.2	Gambaran Umum Aplikasi	26
4.1.3	Identifikasi Aktor.....	27
4.1.4	Analisis Kebutuhan Fungsional.....	27
4.1.4.1	<i>Data flow diagram level 1</i>	29
4.1.4.2	<i>Data flow diagram level 2 : Interaksi dengan pengguna</i>	31
4.1.4.3	<i>Data flow diagram level 2 : Monitor cuaca</i>	34
4.1.5	Analisis Kebutuhan Non-Fungsional.....	34
4.2	Perancangan Perangkat Lunak	35
4.2.1	Perancangan Arsitektur Sistem.....	35
4.2.2	Perancangan Arsitektur Diagram.....	36
4.2.3	Perancangan Basis Data.....	38
4.2.4	Perancangan Antarmuka.....	43
BAB V IMPLEMENTASI DAN PENGUJIAN.....		49
5.1	Implementasi	49



5.1.1	Spesifikasi Sistem.....	50
5.1.1.1	Spesifikasi Perangkat Keras.....	50
5.1.1.2	Spesifikasi Perangkat Lunak.....	50
5.1.2	Batasan-batasan implementasi.....	51
5.1.3	Implementasi Basis Data.....	51
5.1.4	Implementasi Class Dan Assets.....	53
5.1.5	Implementasi Antarmuka.....	55
5.1.6	Implementasi Fitur <i>Mobile</i>	58
5.1.6.1	<i>Geolocation</i>	58
5.1.6.2	Notifikasi Cuaca.....	60
5.2	Pengujian.....	63
5.2.1	Pengujian Unit.....	63
5.2.2	Analisis Hasil Pengujian Unit.....	77
5.2.3	Pengujian Validasi.....	78
5.2.1.1	Kasus Uji Validasi.....	78
5.2.1.2	Hasil Pengujian Validasi.....	88
5.2.4	Analisis Hasil Pengujian Validasi.....	90
5.2.5	Pengujian Kompatibilitas.....	90
5.2.3.1	Pengujian Android Versi 3.0.....	90
5.2.3.2	Pengujian Android Versi 4.0.....	90
5.2.3.3	Pengujian Android Versi 4.3.....	91
5.2.6	Analisis Hasil Pengujian Kompatibilitas.....	91
5.2.7	Pengujian UAT (<i>User Acceptance Test</i>).....	92
5.2.8	Analisis Hasil Pengujian UAT.....	93
BAB VI KESIMPULAN DAN SARAN.....		97
6.1	Kesimpulan.....	97
6.2	Saran.....	98
DAFTAR PUSTAKA.....		99
LAMPIRAN.....		101

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Aplikasi PhoneGap.....	15
Gambar 2.2 Interaksi Perangkat pada Aplikasi PhoneGap.....	15
Gambar 2.3 <i>Waterfall Model</i>	17
Gambar 3.1 Diagram Alir Metodologi Penelitian	20
Gambar 4.1 Diagram Pohon Analisis dan Perancangan.....	24
Gambar 4.2 <i>Data Flow Diagram level 1</i>	30
Gambar 4.3 <i>Data flow diagram level 2</i> : Interaksi dengan pengguna	32
Gambar 4.4 <i>First level factoring</i> : Interaksi dengan pengguna	33
Gambar 4.5 <i>Data flow diagram level 2</i> : Monitor cuaca	34
Gambar 4.6 <i>First level factoring</i> : Monitor cuaca	34
Gambar 4.7 Perancangan Arsitektur Sistem.....	36
Gambar 4.8 Perancangan Arsitektur Diagram	37
Gambar 4.9 <i>Entity Relationship Diagram</i>	42
Gambar 4.10 <i>Site Map</i> Aplikasi	43
Gambar 4.11 Antarmuka Halaman Menu Utama.....	44
Gambar 4.12 Antarmuka Halaman Proses Budidaya	45
Gambar 4.13 Antarmuka Halaman Cara Budidaya.....	46
Gambar 4.14 Antarmuka Halaman Daftar.....	47
Gambar 4.15 Antarmuka Halaman Detail	48
Gambar 5.1 Diagram Pohon Implementasi	49
Gambar 5.2 Diagram <i>Konseptual Entity Relationship</i>	52
Gambar 5.3 Implementasi Query <i>Create</i> tabel budidaya	52
Gambar 5.4 Implementasi Query <i>Create</i> tabel agenda.....	53
Gambar 5.5 Implementasi Query <i>Create</i> tabel ekonomi.....	53
Gambar 5.6 Tampilan Halaman Menu Utama	55
Gambar 5.7 Tampilan Halaman Proses Budidaya.....	56
Gambar 5.8 Tampilan Halaman Cara Budidaya.....	56
Gambar 5.9 Tampilan Halaman Pestisida Nabati.....	57
Gambar 5.10 Tampilan Halaman Detail Pestisida Nabati	58
Gambar 5.11 Implementasi <i>Geolocation</i>	59

Gambar 5.12 Tampilan Informasi Lokasi.....	60
Gambar 5.13 Implementasi Algoritma Notifikasi Cuaca.....	61
Gambar 5.14 Tampilan Notifikasi Cuaca.....	62
Gambar 5.15 Diagram Pohon Pengujian.....	63
Gambar 5.16 Kasus Uji Unit Monitor Cuaca.....	65
Gambar 5.17 <i>Flow Graph</i> Monitor Cuaca.....	66
Gambar 5.18 Hasil Uji Unit Monitor Cuaca Skenario 1.....	67
Gambar 5.19 Hasil Uji Unit Monitor Cuaca Skenario 2.....	68
Gambar 5.20 Hasil Uji Unit Monitor Cuaca Skenario 3.....	69
Gambar 5.21 Hasil Uji Unit Monitor Cuaca Skenario 4.....	69
Gambar 5.22 Kasus Uji Unit Tambah Agenda.....	70
Gambar 5.23 <i>Flow Graph</i> Tambah Agenda.....	71
Gambar 5.24 Hasil Uji Unit Tambah Agenda Skenario 1.....	72
Gambar 5.25 Hasil Uji Unit Tambah Agenda Skenario 1 pada <i>Database</i>	72
Gambar 5.26 Hasil Uji Unit Tambah Agenda Skenario 2.....	73
Gambar 5.27 Hasil Uji Unit Tambah Agenda Skenario 2 pada <i>Database</i>	73
Gambar 5.28 Kasus Uji Unit Tambah Hasil Panen.....	74
Gambar 5.29 <i>Flow Graph</i> Tambah Hasil Panen.....	75
Gambar 5.30 Hasil Uji Unit Tambah Hasil Panen Skenario 1.....	76
Gambar 5.31 Hasil Uji Unit Tambah Hasil Panen Skenario 1 pada <i>Database</i> ...	76
Gambar 5.32 Hasil Uji Unit Tambah Hasil Panen Skenario 2.....	77
Gambar 5.33 Hasil Uji Unit Tambah Hasil Panen Skenario 2 pada <i>Database</i> ...	77

DAFTAR TABEL

Tabel 4.1 Identifikasi Aktor	27
Tabel 4.2 Spesifikasi Kebutuhan Fungsional	27
Tabel 4.3 Spesifikasi Kebutuhan Non-Fungsional	35
Tabel 4.4 Struktur Tabel Fase Budidaya	38
Tabel 4.5 Struktur Tabel Tahap Budidaya	38
Tabel 4.6 Struktur Tabel Proses Budidaya	38
Tabel 4.7 Struktur Tabel Hama	39
Tabel 4.8 Struktur Tabel Pestisida Nabati	39
Tabel 4.9 Struktur Tabel Pupuk Organik	40
Tabel 4.10 Struktur Tabel Agenda	40
Tabel 4.11 Struktur Tabel Ekonomi	40
Tabel 4.12 Struktur Tabel Benih	41
Tabel 4.13 Struktur Tabel Istilah	41
Tabel 4.14 Struktur Tabel Cara Budidaya	41
Tabel 4.15 Keterangan Antarmuka Halaman Menu Utama	44
Tabel 4.16 Keterangan Antarmuka Halaman Proses Budidaya	45
Tabel 4.17 Keterangan Antarmuka Halaman Cara Budidaya	46
Tabel 4.18 Keterangan Antarmuka Halaman Daftar	47
Tabel 4.19 Keterangan Antarmuka Halaman Detail	48
Tabel 5.1 Spesifikasi Perangkat Keras Komputer	50
Tabel 5.2 Spesifikasi Perangkat Keras <i>Smartphone</i>	50
Tabel 5.3 Spesifikasi Perangkat Lunak Komputer	50
Tabel 5.4 Spesifikasi Perangkat Lunak <i>Smartphone</i>	51
Tabel 5.5 Implementasi class pada kode program *.java	53
Tabel 5.6 Implementasi assets pada kode program *.html	54
Tabel 5.7 Implementasi assets pada kode program *.js	54
Tabel 5.8 Kasus uji Validasi Lihat Cara Budidaya	78
Tabel 5.9 Kasus uji Validasi Lihat Hama	79
Tabel 5.10 Kasus uji Validasi Lihat Pestisida Nabati	79
Tabel 5.11 Kasus uji Validasi Lihat Pupuk Organik	80

Tabel 5.12 Kasus uji Validasi Lihat Daftar Agenda.....	80
Tabel 5.13 Kasus uji Validasi Tambah Agenda dengan parameter input sesuai..	80
Tabel 5.14 Kasus uji Validasi Tambah Agenda dengan parameter input kosong	81
Tabel 5.15 Kasus uji Validasi <i>Check</i> Agenda	81
Tabel 5.16 Kasus uji Validasi Hapus Agenda	82
Tabel 5.17 Kasus uji Validasi Lihat Daftar Pengeluaran	82
Tabel 5.18 Kasus uji Validasi Tambah Pengeluaran dengan parameter input sesuai	83
Tabel 5.19 Kasus uji Validasi Tambah Pengeluaran dengan parameter input kosong	83
Tabel 5.20 Kasus uji Validasi Hapus Pengeluaran.....	84
Tabel 5.21 Kasus uji Validasi Lihat Daftar Hasil Panen.....	84
Tabel 5.22 Kasus uji Validasi Tambah Hasil Panen dengan parameter input sesuai	85
Tabel 5.23 Kasus uji Validasi Tambah Hasil Panen dengan parameter input kosong	85
Tabel 5.24 Kasus uji Validasi Hapus Hasil Panen	86
Tabel 5.25 Kasus uji Validasi Lihat Proses Budidaya	86
Tabel 5.26 Kasus uji Validasi Proses Budidaya Selanjutnya	87
Tabel 5.27 Kasus uji Validasi Hapus Hasil Panen	87
Tabel 5.28 Kasus uji Validasi Monitor Cuaca	88
Tabel 5.29 Kasus uji Validasi Lihat Istilah	88
Tabel 5.30 Hasil Pengujian Validasi	89
Tabel 5.31 Kasus Uji dan Hasil Pengujian Kompatibilitas Android Versi 3.0	90
Tabel 5.32 Kasus Uji dan Hasil Pengujian Kompatibilitas Android Versi 4.0	90
Tabel 5.33 Kasus Uji dan Hasil Pengujian Kompatibilitas Android Versi 4.3	91
Tabel 5.34 Hasil Pengujian <i>Usability</i>	92
Tabel 5.35 Interpretasi skor Likert	93
Tabel 5.36 Persentase Keseluruhan Jawaban	93
Tabel 5.37 Status Pengujian UAT	95

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini pola konsumsi organik telah menjadi tren gaya hidup sehat. Hal tersebut dapat dilihat dari pesatnya pertumbuhan pasar produk organik dunia. Dalam 10 tahun terakhir, pasar organik meningkat hingga 228% dengan nilai perdagangan produk organik hingga 59,1 miliar dollar AS [KOM-12]. Selain itu, masyarakat telah menyadari bahaya penggunaan pestisida kimiawi pada pertanian konvensional dapat mengakibatkan keracunan pangan bahkan pencemaran lingkungan. Menurut WHO yang dikutip oleh LESKOFI (Lembaga Studi dan Konsultasi Farmakologi), ditemukan 20.000 orang meninggal karena keracunan pestisida dan sekitar 10.000 orang mengalami dampak pestisida yang berbahaya seperti kanker, cacat, mandul, dan hepatitis setiap tahunnya [PRI-09]. Hal tersebut menjadi salah satu alasan masyarakat untuk beralih ke pola konsumsi organik. Karena budidaya tanaman secara organik adalah teknik budidaya yang mengandalkan bahan-bahan alami tanpa menggunakan bahan kimia sintetis [ANM-10]. Sehingga produk pertanian organik aman dari residu pestisida kimiawi yang berbahaya bagi kesehatan.

Seiring dengan semakin pesatnya pertumbuhan pasar produk organik dunia, maka budidaya pertanian organik perlu diprioritaskan pada tanaman bernilai ekonomis tinggi untuk memenuhi kebutuhan pasar domestik dan ekspor. Di Indonesia, produk hortikultura masih menjadi salah satu produk pertanian unggulan bahkan lebih unggul dibanding negara lain di ASEAN[SIN-14]. Salah satu komoditas hortikultura yang sangat dibutuhkan oleh masyarakat Indonesia adalah cabai. Karena cabai merupakan salah satu bumbu yang banyak digunakan dalam berbagai makanan olahan nusantara. Cabai tidak hanya unggul dengan rasa pedasnya, tetapi juga memiliki kandungan nilai gizi yang tinggi diantaranya protein, karbohidrat, kalsium, vitamin A, vitamin C, betakaroten dan lain-lain. Berbagai manfaat penting cabai bagi tubuh diantaranya sebagai antioksidan,

menurunkan kadar kolesterol, mencegah osteoporosis dan berbagai manfaat lain[NSU-13].

Penggunaan bahan kimia pada pertanian cabai konvensional umumnya dilakukan oleh petani melebihi dosis anjuran. Hal tersebut dipacu oleh kebutuhan pasar demi memaksimalkan hasil panen mengingat pendeknya umur tanaman cabai. Karena cabai merupakan bumbu yang rutin dikonsumsi sehari-hari, sudah seharusnya keamanan produk ini diperhatikan dari residu bahan-bahan kimia. Salah satunya dengan beralih menggunakan produk cabai organik meskipun dengan harga yang relatif lebih mahal daripada produk non-organik. Harga produk organik yang mahal disebabkan oleh budidaya secara organik membutuhkan pengawasan intensif dan lebih kompleks, karena semua kebutuhan harus disiapkan sendiri mulai dari pupuk organik hingga pestisida alami. Daripada membeli produk pertanian organik dengan harga relatif mahal lebih baik melakukan budidaya organik sendiri. Karena budidaya tanaman secara organik selain dilakukan di lahan pertanian yang luas, juga dapat dilakukan di pekarangan rumah yang biasanya dibiarkan tidak produktif.

Dalam melakukan budidaya sendiri, tentunya dibutuhkan pengetahuan lebih tentang cara membuat pupuk organik dan pestisida alami serta memahami tahapan-tahapan budidaya tanaman cabai secara organik. Saat ini telah banyak referensi yang menawarkan cara budidaya organik, petunjuk pembuatan pestisida alami, dan petunjuk pembuatan pupuk organik yang masih dalam bentuk buku. Sehingga, untuk melakukan proses budidaya secara mandiri memerlukan semua buku referensi tersebut agar informasi yang dibutuhkan dapat diperoleh secara lengkap. Dengan membeli semua buku tersebut, tentunya membutuhkan biaya yang tidak sedikit dan pada akhirnya menjadi tidak praktis.

Berkembangnya ilmu pengetahuan tentu tidak lepas dari perkembangan teknologi, salah satunya perkembangan teknologi *mobile*. Selain sebagai alat komunikasi, penggunaan ponsel cerdas juga telah banyak dimanfaatkan untuk tujuan lain dengan memanfaatkan teknologi yang ditawarkan seperti GPS (*Global Positioning System*), *digital compass*, *accelerometer*, *gyroscope* dan lain-lain. Beragam aplikasi ditawarkan untuk memudahkan pengguna dalam memberikan semua informasi secara cepat dengan keakuratan tinggi dengan memanfaatkan

beberapa teknologi mobile yang canggih tersebut. Jadi, memungkinkan semua kebutuhan informasi ditanam kedalam aplikasi yang dikemas secara praktis. Pemanfaatan beberapa teknologi mobile juga dapat dilakukan dalam memperkaya fitur-fitur yang menunjang aplikasi.

Dari uraian diatas, dapat ditarik permasalahan bagaimana masyarakat dapat mengetahui langkah-langkah budidaya cabai secara organik, pembuatan pupuk organik dan pestisida alami, maka dari itu penulis mengangkat judul “**Rancang Bangun Aplikasi *Mobile* Panduan Budidaya Tanaman Cabai Organik Secara Mandiri**” untuk memecahkan masalah tersebut. Aplikasi *mobile* ini akan diimplementasikan pada platform Android dengan memberikan langkah-langkah budidaya tanaman cabai mulai dari persemaian, penanaman, cara perawatan, dan pengendalian hama dengan pestisida alami. Sehingga, aplikasi ini diharapkan dapat memudahkan pengguna dengan memanfaatkan teknologi mobile dalam melakukan budidaya cabai secara organik serta mendapatkan hasil yang optimal.

1.2 Rumusan Masalah

Berdasarkan latar belakang, dirumuskan permasalahan pada penelitian ini sebagai berikut.

1. Bagaimana merancang dan mengimplementasikan aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri yang dapat menjadi acuan dalam melakukan budidaya secara mandiri?
2. Bagaimana menguji aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri yang dapat memudahkan pengguna dalam melakukan budidaya secara mandiri?

1.3 Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian ini dibatasi dalam hal:

1. Aplikasi dibangun pada *platform* Android dengan menggunakan pendekatan pengembangan sistem *hybrid*.
2. Aplikasi dibangun dengan menggunakan bahasa pemrograman HTML, CSS3, dan Javascript.

3. Aplikasi dibangun dengan menggunakan *mobile framework* PhoneGap dan *jQuery mobile*.
4. Penyimpanan data aplikasi dibangun dengan menggunakan SQLite.
5. Pengujian unit dibatasi hanya beberapa fitur, yaitu monitor cuaca, tambah agenda dan tambah hasil panen untuk mengetahui kesesuaian algoritma dari fitur tersebut.
6. Pengujian seluruh fitur pada aplikasi menggunakan pengujian validasi untuk mengetahui kesesuaian dari masing-masing fitur.
7. Pengujian *compatibility* digunakan untuk menguji tampilan dan kesesuaian fungsi dari fitur-fitur pada aplikasi saat diterapkan pada platform android API versi 3.0, 4.0 dan 4.3.
8. Pengujian *usability* yang dilakukan menggunakan pengujian UAT untuk mengukur kemudahan penggunaan dan pencapaian kegunaan berdasarkan konten yang disajikan pada aplikasi.
9. Penerapan teknik budidaya tanaman cabai secara organik dilahan sempit dengan menggunakan media polibag.
10. Pestisida alami yang digunakan adalah pestisida botani yang berasal dari ekstrak tanaman.

1.4 Tujuan

Tujuan dari penelitian ini untuk membuat aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri yang dapat menjadi acuan dalam memudahkan pengguna untuk melakukan budidaya secara mandiri.

1.5 Manfaat

Manfaat yang bisa didapat dari penelitian ini adalah :

1. Masyarakat dapat melakukan budidaya cabai organik secara mandiri dengan memanfaatkan aplikasi *mobile* budidaya tanaman cabai sebagai pemandu.
2. Menjadikan lahan pekarangan lebih produktif, karena dalam proses budidaya menggunakan media polibag yang dapat diterapkan di lahan pekarangan.
3. Memberikan dampak positif bagi lingkungan, karena dalam proses budidaya tanaman tanpa penggunaan pupuk kimia maupun pestisida kimia.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini sebagai berikut :

BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini menjelaskan pustaka yang dibutuhkan baik mengenai *software* maupun *hardware* serta teori yang diperlukan dalam perancangan aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri.

BAB III METODE PENELITIAN

Dalam bab ini akan dibahas mengenai metodologi untuk perancangan aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri.

BAB IV ANALISIS DAN PERANCANGAN

Dalam bab ini akan dibahas tahap analisa kebutuhan dan perancangan aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri.

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dan hasil pengujian fitur dari aspek fungsional maupun non-fungsional dari aplikasi. Kemudian dilakukan analisa terhadap hasil pengujian.

BAB VI KESIMPULAN DAN SARAN

Bab ini meliputi kesimpulan dari apa yang dijelaskan pada bab-bab sebelumnya serta saran yang dapat dijadikan masukan dalam penelitian lebih lanjut.

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan dibahas mengenai kajian pustaka dan dasar teori yang digunakan penulis dalam menyusun skripsi. Dasar teori digunakan sebagai dasar pembuatan perangkat lunak dan dasar informasi tentang objek penelitian seperti cabai, pertanian organik, teknik budidaya, pupuk serta pestisida organik.

2.1 Cabai

Cabai (*Capsicum sp*) adalah tanaman tahunan dengan tinggi mencapai satu meter, merupakan tumbuhan perdu yang berkayu, buahnya berasa pedas, dan tumbuh di daerah dengan iklim tropis. Secara umum cabai bisa ditanam pada area persawahan maupun tegalan pada daerah dataran rendah maupun dataran tinggi baik pada musim kemarau atau musim penghujan. Akan tetapi, tanaman ini akan tumbuh baik di lahan dataran rendah yang tanahnya gembur dan kaya bahan organik, tekstur ringan sampai sedang, pH tanah 5,5-6,8, drainase baik, dan cukup tersedia unsur hara bagi pertumbuhan tanaman[ASA-99].

2.1.1 Cabai Rawit

Cabai rawit (*Capsicum frutescens*) merupakan tanaman berumur panjang yang termasuk dalam famili Solanaceae. Apabila dirawat dengan baik dan kebutuhan haranya tercukupi, tanaman ini dapat hidup sampai 2-3 tahun. Terdapat beberapa macam cabai rawit antara lain rawit kecil, sedang dan besar. Umumnya cabai rawit kecil rasanya sangat pedas. Cabai rawit digunakan untuk sayur, bumbu masak, asinan dan obat.

Cabai rawit dapat ditanam di dataran rendah maupun di dataran tinggi, namun tanaman ini lebih cocok ditanam pada ketinggian antara 0-500 m dpl. Karena produksi pada ketinggian diatas 500 m dpl berpengaruh pada waktu panennya yang lebih panjang. Tanah yang baik untuk pertanaman cabai rawit adalah tanah gembur yang kaya akan bahan organik dan pH tanah netral antara 6-7[BCR-14].

2.1.2 Cabai Merah

Cabai merah (*Capsicum annum*) termasuk famili Solanaceae dan merupakan salah satu komoditas sayuran yang memiliki banyak manfaat, bernilai ekonomi tinggi dan mempunyai prospek pasar yang menarik. Selain dapat dikonsumsi segar untuk campuran bumbu masak juga dapat diawetkan misalnya dalam bentuk acar, saus, tepung cabai dan buah kering.

Cabai merah dapat ditanam di dataran rendah maupun dataran tinggi, namun tanaman ini lebih cocok ditanam pada ketinggian antara 0–1000 m dpl. Tanah yang baik untuk pertanaman cabai merah ini adalah tanah gembur, subur, kaya akan bahan organik, dan pH tanah antara 6–7. Kandungan air tanah juga perlu diperhatikan yang berhubungan dengan tempat tumbuh tanaman cabai tersebut di sawah atau tegalan[BCM-14].

2.2 Pertanian Organik

Pertanian organik adalah teknik budidaya pertanian yang mengandalkan bahan-bahan alami tanpa menggunakan bahan-bahan kimia sintetis. Tujuan utama pertanian organik adalah menyediakan produk-produk pertanian, terutama bahan pangan yang aman bagi kesehatan produsen dan konsumennya serta tidak merusak lingkungan. Gaya hidup sehat demikian telah melembaga secara internasional yang mensyaratkan jaminan bahwa produk pertanian harus beratribut aman dikonsumsi (*food safety attributes*), kandungan nutrisi tinggi (*nutritional attributes*) dan ramah lingkungan (*eco-labelling attributes*). Preferensi konsumen seperti ini menyebabkan permintaan produk pertanian organik dunia meningkat pesat[ANM-10].

Pertanian organik merupakan kegiatan bercocok tanam yang akrab dengan lingkungan. Pertanian organik berusaha meminimalkan dampak negatif bagi alam sekitar. Ciri utama pertanian organik adalah pengguna varietas lokal yang masih alami, diikuti dengan penggunaan pupuk organik dan pestisida organik[ADK-02].

2.3 Teknik Budidaya Cabai Secara Organik

Budidaya cabai secara organik adalah usaha budidaya tanaman cabai yang dalam pelaksanaannya meminimalkan penggunaan senyawa-senyawa kimia, seperti pupuk dan pestisida kimia. Teknik budidaya secara organik ini mengoptimalkan

penggunaan bahan-bahan organik sebagai sumber nutrisi bagi pertumbuhan dan bahan-bahan hayati dalam mengendalikan hama. Tahapan-tahapan teknik budidaya secara organik dalam polibag sebagai berikut [NSU-13].

1. Persiapan awal

Pada fase ini dilakukan pemilihan benih yang akan digunakan dalam proses budidaya. Kemudian membuat media tanam dengan mencampur tanah dan pupuk organik untuk dimasukkan ke dalam plastik semai. Setelah itu benih cabai siap ditanam pada plastik semai satu persatu.

2. Perawatan semai

Pada fase ini dilakukan perawatan dengan melakukan penyiraman rutin setiap hari hingga bibit cabai memiliki daun 4-6 helai yang menandakan bibit siap ditanam.

3. Penanaman

Pada fase ini dilakukan pembuatan media tanam dengan mencampur tanah, sekam padi dan pupuk organik untuk dimasukkan ke dalam polibag. Kemudian bibit cabai yang telah siap tanam dipindahkan ke dalam polibag secara hati-hati tepat ditengah. Padatkan media tanam hingga bibit cabai dapat berdiri dengan kuat. Setelah itu, pasang ajir bambu sebagai penahan bibit.

4. Pemeliharaan

Pada fase ini berisi kegiatan pemeliharaan diantaranya penyiraman, pemupukan, penyiangan gulma, dan pengendalian hama. Perawatan dilakukan hingga tanaman cabai tidak produktif lagi.

2.4 Pupuk Organik

Kompos adalah pupuk organik yang bahan dasarnya dari pelapukan bahan tanaman atau limbah organik. Bahan dasar yang biasa digunakan seperti jerami, sekam, rumput-rumputan, sampah kota atau limbah pabrik. Proses pengomposan adalah suatu proses mikrobiologi. Bahan organik dirombak oleh aktifitas mikroorganisme sehingga dihasilkan energi dan unsur karbon sebagai pembangun sel-sel tumbuh. Sumber energi diperoleh dari unsur N pada bahan organik mentah. Di dalam pengomposan akan terjadi perubahan yang dilakukan

oleh mikroorganisme, yaitu berupa penguraian selulose, hemiselulose serta lainnya menjadi karbondioksida (CO₂) dan air, pengikatan unsur hara oleh mikroorganisme yang akan dilepaskan kembali bila mikroorganisme mati, serta pembebasan unsur hara senyawa organik menjadi senyawa anorganik yang akan tersedia bagi tanaman[MUS-03].

2.5 Pestisida Alami

Pestisida alami berasal dari bahan-bahan yang terdapat di alam yang diekstraksi, diproses, atau dibuat menjadi konsentrat dengan tidak mengubah struktur kimianya. Pestisida alami dapat dikelompokkan menjadi 3 golongan sebagai berikut[NOV-02].

1. Pestisida botani yang berasal dari ekstrak tanaman. Seperti diketahui, berbagai jenis tanaman memproduksi senyawa kimia untuk melindungi dirinya sendiri dari serangan OPT (organisme pengganggu tanaman). Senyawa inilah yang kemudian diambil dan dipakai untuk melindungi tanaman lain.
2. Pestisida biologis yang mengandung mikroorganisme pengganggu, seperti bakteri patogenik, virus, dan jamur. Mikroorganisme ini secara alami memang merupakan musuh OPT, yang kemudian dikembang-biakkan untuk keperluan perlindungan tanaman. Proses manufaktur dari organisme ini telah memungkinkan petani memekainya sebagaimana memakai pestisida lainnya dengan cara menyemprot atau menebarkannya.
3. Pestisida berbahan dasar mineral anorganik yang terdapat pada kulit bumi. Biasanya bahan mineral ini berbentuk kristal, tidak mudah menguap, dan bersifat stabil secara kimia, seperti belerang dan kapur. Minyak bumi atau minyak nabati dan sabun pun dapat dipakai untuk mengendalikan OPT. Pada pertanian organik, minyak dan sabun sangat lazim dipakai.

Ketiga jenis pestisida tersebut berlabel ramah lingkungan yang mengacu pada dua hal. Pertama, residu pestisida alami lebih cepat terurai oleh komponen-komponen alam, sehingga tidak akan menyebabkan pencemaran air dan tanah. Kedua, daya racun dari pestisida alami bersifat selektif. Artinya, pestisida alami

hanya mematikan OPT jenis tertentu dan relatif aman bagi musuh alami, manusia, mamalia, dan ikan.

2.6 Android

Android adalah salah satu sistem operasi hasil pengembang vendor Linux yang dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer tablet. Keberadaan Android pertama kali dikembangkan oleh vendor Android, Inc. dengan dukungan finansial dari Google. Setelah beberapa waktu Google membiayai pengembangan Android, pada tahun 2005 Google memutuskan untuk membeli keberadaan Android dari vendor Android, Inc. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance (OHA), konsorsium (Kerjasama pembiayaan) dari berbagai perusahaan, yang tercatat 34 perusahaan terlibat dalam pengembangan proyek besar ini, seperti: perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia yang bertujuan untuk memajukan standar umum perangkat-perangkat seluler[AIG-13].

Sejarah perkembangan versi Android diawali dengan dirilisnya Android beta pada bulan November 2007. Yang merupakan versi komersial pertama, Android 1.0, dirilis pada September 2008. Sejak April 2009, versi Android mulai dikembangkan dengan nama kode-kode yang dinamai berdasarkan makanan ringan dan roti cemilan manis. Masing-masing versi dirilis sesuai urutan alfabet, yakni versi Cupcake (1.5), versi Donut (1.6), versi Eclair (2.0–2.1), Froyo (2.2–2.2.3), Gingerbread (2.3–2.3.7), Honeycomb (3.0–3.2.6), Ice Cream Sandwich (4.0–4.0.4), versi Jelly Bean (4.1–4.3), dan versi KitKat (4.4+).

2.7 *HyperText Markup Language* (HTML)

HyperText Markup Language adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman *web* dan ditampilkan berbagai informasi dalam *web browser*. HTML dibuat oleh Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN yang merupakan lembaga penelitian fisika energi tinggi Jenewa pada tahun 1989[WEN-06]. Dalam dokumen HTML, sebuah tulisan bisa memuat instruksi yang ditandai dengan kode atau biasa disebut TAG. HTML

tidak membedakan penggunaan huruf besar ataupun huruf kecil dari suatu elemen. Suatu elemen HTML terdiri dari beberapa tag beserta teks yang ada dalam tag tersebut. Tag tersebut dinyatakan dengan tanda lebih kecil (<) dan tanda lebih besar (>) serta tag penutup diakhiri dengan tanda garis miring (/) setelah tanda lebih kecil (<). Misalnya, <nama tag> teks yang ditampilkan </nama tag>[AND-01].

2.8 Cascading Style Sheet (CSS)

Cascading Style Sheet adalah suatu bahasa stylesheet yang digunakan untuk mengatur *style* suatu dokumen. Biasanya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML. CSS menggunakan kode-kode yang tersusun untuk menetapkan *style* pada elemen HTML atau dapat juga digunakan untuk membuat style baru yang biasa disebut class. CSS dapat mengubah ukuran teks, mengganti warna *background* pada sebuah halaman, mengubah warna border pada tabel, dan lain-lain[SRR-08].

Terdapat tiga jenis cara penggunaan CSS, yaitu

1. External Style Sheet

External Style Sheet adalah CSS yang dibuat dalam file terpisah dengan ekstensi file .css. Untuk memanggilnya menggunakan script html yang disisipkan antara tag head sebagai berikut.

```
<link rel="stylesheet" type="text/css" href="direktori/filecss.css" />
```

Isi pada bagian href mengarah ke direktori tempat file css. Penggunaan sistem embeding css dengan cara ini sangat disarankan karena mudah untuk melakukan pengeditan, mengatur semantik dan praktis terkumpul menjadi satu file.

2. Internal Style Sheet

Internal Style Sheet adalah cara embeding css dengan menulis langsung didalam file html yang ingin diatur tampilannya. Penulisan ini disisipkan diantara tag head juga dengan diapit oleh tag style sebagai berikut.

```
<style type="text/css">  
body{  
background:#cccccc;
```

```
font-family:Arial;  
}  
</style>
```

3. Inline Style Sheet

Inline Style Sheet adalah penulisan script css langsung pada tag html dengan menambahkan style didalamnya, misalnya :

```
<p style="font-size:20px;">Tulisan yang diatur </p>
```

2.9 Javascript

Javascript adalah bahasa *scripting* yang populer di internet dan dapat bekerja disebagian besar browser populer seperti Internet Explorer, Mozilla Firefox, Netscape dan Opera, Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag SCRIPT[ANS-07]. JavaScript adalah dibuat untuk berinteraksi dengan halaman web dengan 3 bagian sebagai berikut[ZAK-12].

1. ECMAScript, yang didefinisikan dalam ECMA-262 dan menjadi bagian utama dalam menjalankan fungsionalitas.
2. Document Object Model (DOM), yang menyediakan method dan interface untuk bekerja dengan isi dari halaman website.
3. Browser Object Model (BOM), yang menyediakan method dan interfaces untuk berinteraksi dengan browser.

2.10 JQuery

JQuery merupakan library dari javascript yang fokus dalam melakukan *query* objek javascript. *Selector* yang digunakan dalam jquery biasanya mengakses kelas CSS untuk menangkap DOM dan mengolahnya dengan method tertentu[DAV-11]. Beberapa fitur yang menjadi inti dari jquery sebagai berikut.

1. Syntax ekspresif (CSS selectors) untuk menunjuk element pada dokumen.
2. Query yang efektif dan efisien dalam menemukan element pada dokumen.
3. Kumpulan method yang berguna untuk memanipulasi element yang terpilih.
4. Succinct idiom (method berantai) untuk membuat *sequence* dari operasi yang dibuat.

2.11 JQuery Mobile

JQuery Mobile merupakan suatu *framework* web yang dikembangkan oleh tim proyek dari jQuery. Pengembangan difokuskan pada pembuatan suatu *framework* yang sesuai dengan berbagai macam *smartphone* dan *tablet*. Framework jQuery Mobile memiliki kompatibilitas dengan framework aplikasi *mobile* lainnya seperti PhoneGap, Worklight dan lainnya. JQuery Mobile menyediakan komponen UI widget seperti button, listview, header dan elemen form dan navigasi [JQM-14].

Beberapa fitur dari jQuery Mobile yaitu:

1. Kompatibel dengan semua *platform mobile* utama serta semua *desktop browser* utama, termasuk iOS, Android, Blackberry, WebOS, Symbian, Windows Phone, dan lainnya.
2. *Theming framework* yang memungkinkan pembuatan tema sesuai keinginan.
3. Dependensi terbatas dan ringan untuk mengoptimalkan kecepatan.
4. Basis kode yang sama secara otomatis akan menyesuaikan untuk skala layar manapun.
5. HTML5 yang berbasis konfigurasi untuk membuat halaman dengan *scripting* minimal.

2.12 JSON

JSON (*JavaScript Object Notation*) pertama kali dikenalkan oleh Douglas Crockford adalah bagian dari javascript, dibuat dengan menggunakan pola dari javascript untuk merepresentasikan data terstruktur. Bagian terpenting dari JSON adalah format data bukan bahasa pemrogramannya. JSON merupakan data format, jadi JSON memiliki *parsers* dan *serializers* yang ada dalam banyak bahasa pemrograman[ZAK-12].

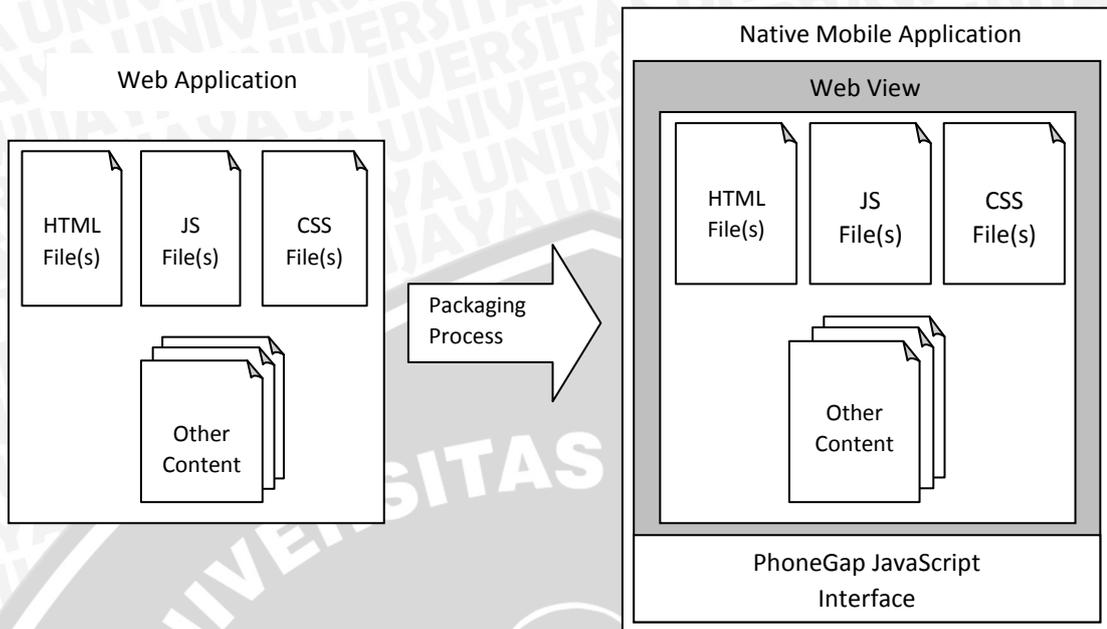
2.13 SQLite

SQLite adalah Relational Database Management Server (RDBMS) alternatif yang bersifat *portable* atau tidak memerlukan proses instalasi, dan didukung oleh banyak bahasa pemrograman[DIK-07]. Beberapa fitur dari SQLite sebagai berikut[DEV-11].

1. *Serverless*, SQLite tidak memerlukan proses pada server atau sistem untuk menjalankannya, melainkan hanya sebuah file yang diakses oleh *library* SQLite.
2. *Cross Platform*, semua instan database berada dalam sebuah file yang cross-platform, sehingga tidak memerlukan administrasi.
3. *Self-Contained*, sebuah *library* mengandung keseluruhan dari sistem database, yang langsung terintegrasi pada sebuah aplikasi program
4. *Small Runtime Footprint*, untuk membangun database SQLite hanya membutuhkan kurang dari satu megabyte *library* (kode program) dan hanya membutuhkan beberapa megabyte *memory*, bahkan dengan beberapa adjustment baik ukuran *library* maupun *memory* dapat diperkecil.
5. *Transactional*, SQLite transaction memperbolehkan aksi penyimpanan melalui beberapa proses *thread*.
6. *Full Featured*, SQLite mendukung hampir sebagai besar standar SQL92 (SQL2).
7. *Highly Reliable*, Tim pengembang SQLite mengembangkan melalui kode program yang sangat serius serta telah melewati proses testing.

2.14 PhoneGap

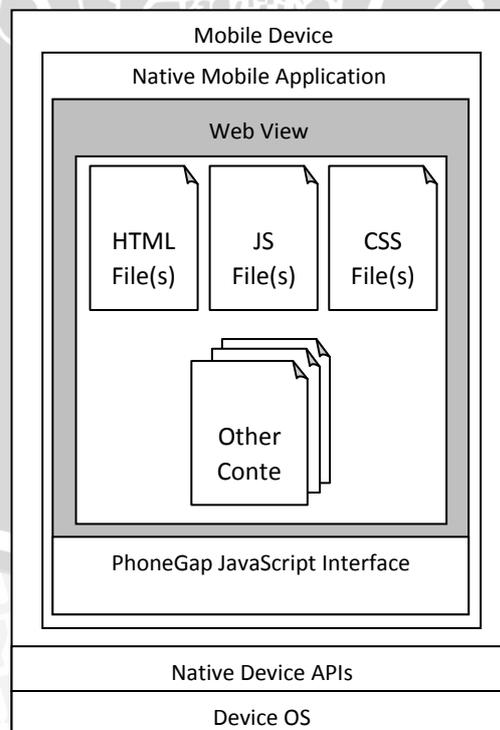
PhoneGap merupakan suatu *framework* produksi Nitobi yang digunakan dalam mengembangkan aplikasi perangkat bergerak. Nitobi sendiri telah diakuisisi oleh Adobe Systems pada tahun 2011. PhoneGap memungkinkan programmer dapat membangun aplikasi untuk perangkat mobile menggunakan JavaScript, HTML5, dan CSS3 yang bukan merupakan bahasa spesifik suatu perangkat seperti Objective-C. Aplikasi yang dihasilkan adalah *hybrid*, yang berarti bahwa aplikasi tersebut bukan *native* murni karena seluruh pembuatan *layout* dilakukan melalui tampilan web dan bukan *framework* UI asli dari platform yang spesifik. Dan *hybrid* bukan juga aplikasi murni berbasis web karena aplikasi tersebut tidak hanya aplikasi *web*, tetapi dikemas sebagai aplikasi untuk distribusi dan memiliki akses ke API (*Application Programming Interface*) dari perangkat *native*[PHG-14]. PhoneGap bekerja dengan cara merubah *web application package* menjadi *native application* seperti pada Gambar 2.1[WRG-12].



Gambar 2.1 Arsitektur Aplikasi PhoneGap

sumber : [WRG-12]

Aplikasi yang telah dibuat akan ditampilkan dalam bentuk *web view* yang memungkinkan pengguna untuk melakukan interaksi dengan aplikasi seperti pada Gambar 2.2[WRG-12].



Gambar 2.2 Interaksi Perangkat pada Aplikasi PhoneGap

sumber : [WRG-12]

Aplikasi yang telah selesai dibuat dengan PhoneGap akan dirubah dengan Software Development Kit (SDK) platform yang didukung oleh PhoneGap menjadi *native application*. Pada saat ini PhoneGap mendukung pengembangan beberapa platform yaitu Apple iOS, BlackBerry, Google Android, LG webOS, Microsoft Windows Phone, Nokia Symbian OS, Tizen, Bada, Firefox OS, dan Ubuntu Touch[WRG-12]. Dalam penyimpanan data PhoneGap menggunakan teknologi *local storage* yang terdapat pada HTML5 selain itu juga PhoneGap dapat menggunakan *API storage* untuk mengakses SQLite pada perangkat berbasis Android.

2.15 Aptana Studio

Aptana studio merupakan aplikasi desain dan pengembangan web yang mendukung berbagai macam jenis bahasa pemrograman web seperti ASP, ASP.NET, JSP, PHP, dan lain-lain. Aptana studio dapat berjalan pada banyak sistem operasi diantaranya Linux, Microsoft Windows, dan Mac OS. Aptana Studio menyediakan *editor WYSIWYG visual* atau disebut juga *Design view* dan kode *editor* dengan fitur *syntax highlighting*, *code completion*, dan *code collapsing* serta fitur canggih seperti *real-time syntax checking* dan *code introspection*.

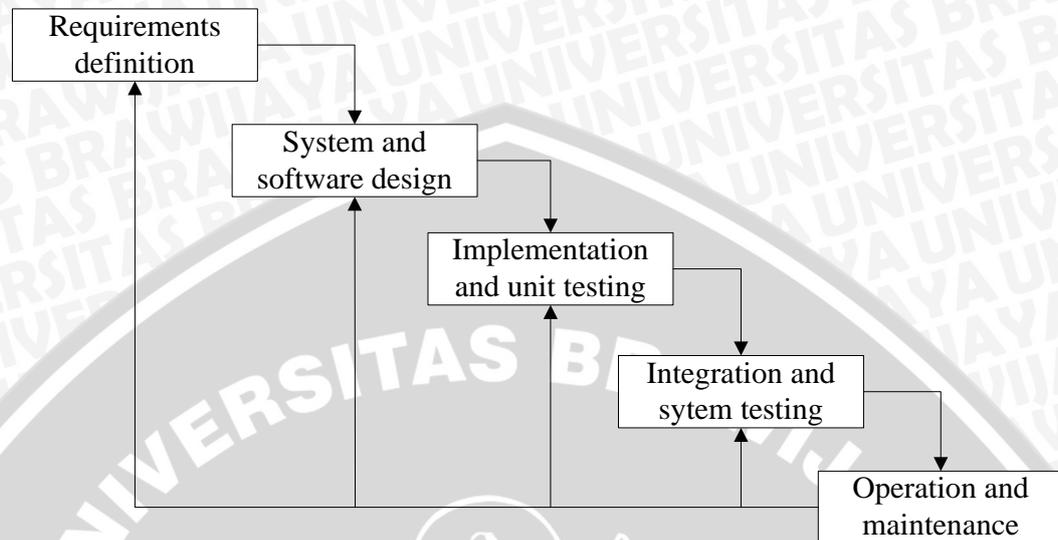
2.16 XAMPP

XAMPP singkatan dari X yang berarti *cross-platform*, Apache, MySQL, PHP, Perl yang merupakan sebuah *web-server* yang mudah digunakan, bersifat gratis dan dapat disebarluaskan[APA-14]. XAMPP digunakan oleh para pengembang dan programmer web untuk menguji dan memvalidasi sebuah halaman *web* tanpa harus terhubung dengan internet secara offline melalui akses ke *localhost*. XAMPP tersedia untuk Microsoft Windows, Linux, Mac OS dan Sun Solaris.

2.17 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan disiplin ilmu dibidang rekayasa yang membahas semua aspek dari seluruh proses pembuatan perangkat lunak dari tahap awal kebutuhan sistem, perawatan sistem, hingga sistem dapat digunakan[SOM-

11]. Pada tahap pembuatan aplikasi dalam skripsi ini menggunakan *waterfall* model yang ditunjukkan pada Gambar 2.3 berikut.



Gambar 2.3 *Waterfall Model*

sumber : [SOM-07]

Tahapan utama dari *waterfall model* tersebut sebagai berikut.

2.17.1 Analisis Kebutuhan (*Requirement Analysis*)

Kebutuhan dari suatu sistem merupakan deskripsi tentang apa yang harus sistem lakukan. Kebutuhan tersebut merefleksikan apa yang dibutuhkan pengguna pada sebuah sistem yang menyediakan suatu tujuan pasti seperti mengontrol device, menempatkan pesanan, atau untuk mencari informasi. Proses pencarian, analisis, dokumentasi dan pengecekan layanan dan keterbatasan disebut *requirement engineering* [SOM-07]. Kebutuhan sistem perangkat lunak sering diklasifikasikan sebagai berikut[SOM-07].

1. Kebutuhan fungsional

Hal ini merupakan penjelasan mengenai layanan yang harus disediakan oleh sistem. Bagaimana sistem merespon input-input tertentu, dan bagaimana merespon dalam kondisi tertentu.

2. Kebutuhan non-fungsional

Kebutuhan non-fungsional merupakan batasan dalam layanan atau fungsi yang di tawarkan oleh sistem. Hal tersebut diantaranya batasan waktu, batasan proses pengembangan, dan batasan standar kemampuan.

2.17.2 Perancangan (Design)

Desain perangkat lunak merupakan proses mengidentifikasi komponen perangkat lunak dan setiap relasi yang terkait di dalamnya berdasarkan kebutuhan pengguna. Ketika mengembangkan sebuah aplikasi, proses desain menjadi diperhatikan tentang bagaimana menggunakan konfigurasi fitur dari sebuah sistem untuk menyalurkan kebutuhan dari sistem [SOM-07]. Pada tahap perancangan dalam skripsi ini, digunakan pemodelan dengan menggunakan satu macam diagram, yaitu *activity diagram*. *Activity diagram* menggambarkan tingkah laku sistem yang dinamis atau bagian dari sistem yang melewati aliran kontrol antara aksi yang dilakukan oleh sistem. *Activity diagram* hampir sama seperti flowchart tetapi *activity diagram* dapat memperlihatkan suatu aliran proses dalam waktu yang sama [PRE-01].

2.17.3 Implementasi

Rekayasa perangkat lunak meliputi semua aktivitas dalam proses pengembangan perangkat lunak dari proses awal analisa kebutuhan sampai proses perawatan dan manajemen dari sistem yang dikembangkan. Tahap kritis dari proses ini tentunya proses implementasi sistem, dimana terdapat proses pembuatan versi perangkat lunak yang bisa dijalankan. Implementasi bisa jadi meliputi proses pengembangan program menggunakan bahasa *high-level* atau *low-level* untuk memenuhi kebutuhan khusus dari perangkat lunak [SOM-07].

2.17.4 Pengujian

Pengujian perangkat lunak dilakukan untuk memastikan bahwa semua persyaratan perangkat lunak telah terpenuhi. Pengujian perangkat lunak memerlukan perancangan kasus uji (*test case*) agar dapat menemukan kesalahan dalam waktu singkat dan usaha minimum. Berbagai macam metode perancangan telah menyediakan mekanisme yang dapat membantu memastikan kelengkapan dari pengujian dan menyediakan kemungkinan tertinggi untuk menemukan

kesalahan-kesalahan dalam perangkat lunak. Terdapat 2 metode pengujian yang digunakan sebagai berikut[PRE-01].

1. *Black-Box Testing*

Black-box testing berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *black-box* merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori berikut.

- Fungsi-fungsi yang tidak benar atau hilang.
- Kesalahan interface.
- Kesalahan dalam struktur data atau akses basis data eksternal.
- Kesalahan kinerja.
- Inisialisasi dan kesalahan terminasi.

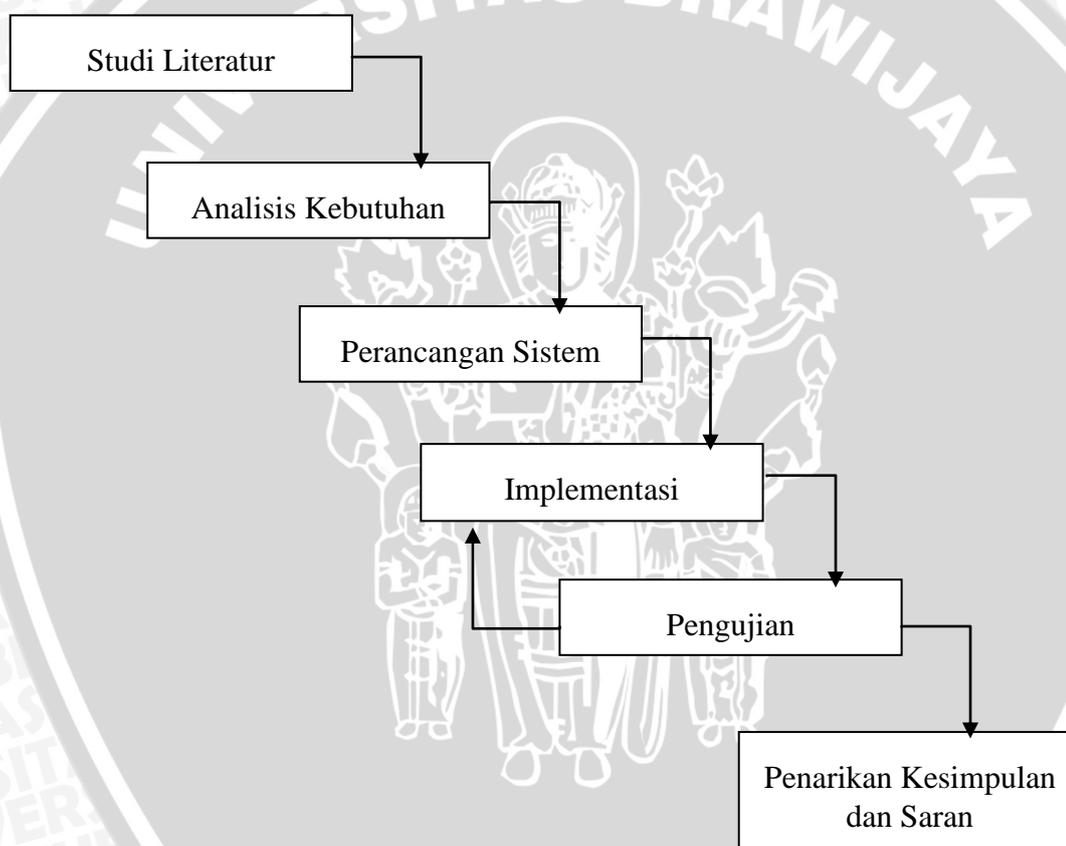
2. *White-Box Testing*

White-box testing merupakan sebuah metode perancangan kasus uji yang menggunakan struktur kontrol dari perancangan prosedural untuk memperoleh kasus uji

BAB III

METODE PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang dilakukan dalam proses pembuatan aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri. Diagram alir dari metode penelitian dalam skripsi ini menggunakan SDLC (*Systems development life cycle*) dengan pendekatan *waterfall model* yang dapat dilihat pada Gambar 3.1 berikut.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Studi Literatur

Studi Literatur adalah metode yang dilakukan dengan mengumpulkan dan mempelajari data-data dari berbagai literatur yang dapat berasal dari buku, jurnal dan dokumentasi dari internet yang berkaitan dengan teori yang akan diterapkan

dalam penelitian ini. Hal tersebut diharapkan dapat memberikan gambaran tentang langkah-langkah yang dapat diterapkan. Beberapa hal yang menjadi bahan studi literatur dalam penelitian ini sebagai berikut:

1. Studi literatur yang berkaitan dengan objek penelitian yaitu cabai, pertanian organik, pupuk organik, dan pestisida alami.
2. Studi literatur yang berkaitan dengan bahasa pemrograman dan *framework* yaitu *hypertext markup language (HTML)*, *cascading style sheets (CSS)*, javascript, jQuery, jQuery mobile, JSON, SQLite, dan phoneGap
3. Studi literatur yang berkaitan dengan pembuatan dan pengembangan aplikasi yaitu aptana studio 3 dan XAMPP.

3.2 Analisis Kebutuhan

Tahap ini dilakukan dengan memanfaatkan tingkah laku dari pengguna sistem yang telah dipelajari pada tahap studi literatur. Informasi panduan budidaya cabai organik diperoleh dari buku “Budidaya Cabai di Lahan Sempit”[NSU-13] dan “Cabe Hibrida Unggul di Lahan dan Polibag”[SAL-13]. Dari informasi yang didapatkan tersebut kemudian ditentukan hal-hal apa saja yang akan diterapkan pada sistem. Sehingga sistem yang dibuat akan menyajikan semua informasi secara lengkap dan fitur penunjang yang tepat. Pemodelan pada sistem menggunakan pendekatan terstruktur, sehingga analisis kebutuhan fungsional sistem digambarkan menggunakan *Data Flow Diagram (DFD)* yang berorientasi pada fungsi. Dan analisis kebutuhan non-fungsional digunakan untuk menganalisa kemampuan sistem agar dapat berjalan dengan baik saat digunakan.

3.3 Perancangan Sistem

Tahap perancangan sistem didasarkan pada hasil analisis kebutuhan sistem yang telah dilakukan. Proses perancangan ini akan dibagi menjadi empat tahapan yaitu:

- a. Perancangan arsitektur sistem

Perancangan arsitektur sistem dilakukan dengan merepresentasikan relasi dari masing-masing komponen penyusun sistem. Beberapa komponen tersebut adalah perangkat *mobile* sebagai media dari sistem, kode program

yang terdiri dari HTML, CSS, dan Javascript, serta *framework* yang terdiri dari JQuery Mobile dan PhoneGap.

b. Perancangan arsitektur diagram

Perancangan arsitektur diagram merepresentasikan fungsi-fungsi yang digunakan dalam aplikasi yang didasarkan dari penggabungan *first level factoring* yang didefinisikan setelah pembuatan *data flow diagram*.

c. Perancangan basis data

Perancangan basis data dilakukan dengan merepresentasikan struktur penyimpanan data yang digunakan oleh sistem. Model representasi dari basis data menggunakan *Entity Relationship Diagram* (ERD) yang dapat menjelaskan seluruh entitas dan atribut yang terlibat beserta relasinya. Setelah rancangan ERD selesai dirancang kemudian dilanjutkan dengan menyusun struktur dari masing-masing tabel.

d. Perancangan antarmuka

Perancangan desain antarmuka dilakukan dengan membuat *layout* dari masing-masing menu yang ditawarkan oleh aplikasi. *Layout* tersebut berisi komponen-komponen seperti *header*, *content*, dan *button* yang akan ditampilkan pada aplikasi.

3.4 Implementasi

Sistem akan dibangun menggunakan bahasa pemrograman HTML, CSS dan JavaScript dengan memanfaatkan beberapa perangkat lunak yaitu:

- a. *Aptana Studio 3* : Sebagai perangkat lunak yang digunakan untuk membangun dan mengembangkan aplikasi.
- b. *XAMPP 1.7.7* : Sebagai perangkat lunak yang digunakan untuk menghubungkan aplikasi agar dapat diakses pada *localhost*.
- c. *jQuery 1.9.0* : Sebagai *framework* yang digunakan dalam mengembangkan aplikasi
- d. *jQuery Mobile 1.4.0* : Sebagai *framework* yang digunakan dalam mengembangkan aplikasi

- e. *PhoneGap* : Sebagai *framework* untuk aplikasi *mobile* secara *cross platform* dengan menggunakan bahasa HTML, CSS dan Javascript.

3.5 Pengujian

Proses pengujian perangkat lunak dilakukan untuk menjamin aplikasi yang dibangun dapat berjalan dengan baik sesuai dengan fungsi dan tujuan saat digunakan. Strategi pengujian yang akan digunakan yaitu pengujian unit (*unit testing*), pengujian validasi (*validation testing*), pengujian kompatibilitas, dan pengujian UAT (*User Acceptance Test*). Pada tahap pengujian unit digunakan metode *white-box testing*, sedangkan tahap pengujian validasi dan kompatibilitas digunakan metode *black-box testing*. Pengujian kompatibilitas dilakukan untuk mengetahui kemampuan aplikasi dalam menampilkan fitur yang sesuai saat berjalan pada versi Android yang berbeda. Sedangkan pengujian UAT dilakukan untuk melihat kebutuhan *usability* pada sistem apakah dapat diterima oleh pengguna dengan baik. Setelah seluruh proses pengujian selesai, maka dilakukan analisis berdasarkan hasil pengujian yang didapatkan.

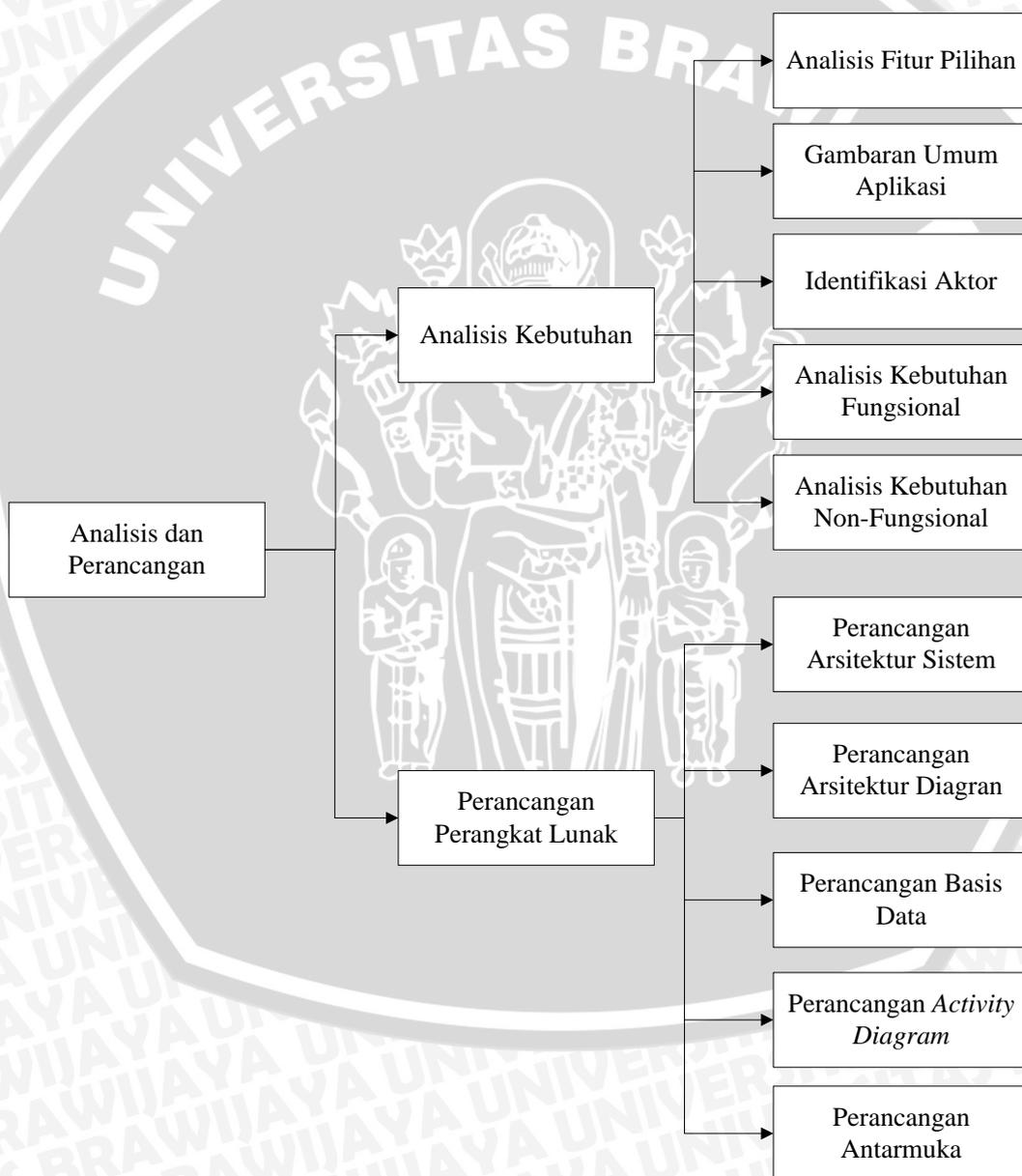
3.6 Pengambilan Kesimpulan dan Saran

Pada tahap terakhir akan dilakukan proses pengambilan kesimpulan dan saran. Pengambilan kesimpulan didapat dari hasil analisis dari pengujian yang telah dilakukan dan saran didapat dari kesimpulan dengan tujuan untuk menyempurnakan kekurangan dari penelitian yang dilakukan.

BAB IV

ANALISIS DAN PERANCANGAN

Pada bab ini dijelaskan mengenai proses perancangan yang dilakukan meliputi dua tahap yaitu analisis kebutuhan dan perancangan perangkat lunak. Tahapan proses analisis dan perancangan ditunjukkan dalam bentuk diagram pohon pada Gambar 4.1 berikut.



Gambar 4.1 Diagram Pohon Analisis dan Perancangan

4.1 Analisis Kebutuhan

Proses analisis kebutuhan ini diawali dengan analisis fitur pilihan, gambaran umum aplikasi, identifikasi aktor, penjabaran tentang kebutuhan fungsional yang akan dimodelkan dalam bentuk *data flow diagram* dan kebutuhan non-fungsional. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

4.1.1 Analisis Fitur Pilihan

Analisis fitur pilihan dilakukan untuk menentukan fitur-fitur apa saja yang akan diterapkan pada aplikasi. Dalam melakukan budidaya cabai organik secara mandiri tentunya dibutuhkan pengetahuan tentang tahapan-tahapan proses budidaya. Informasi tahapan tersebut diperoleh berdasarkan referensi dari buku “Budidaya Cabai di Lahan Sempit”[NSU-13]. Selain itu, pengetahuan tentang pupuk organik juga diperlukan untuk memenuhi kebutuhan hara dan nutrisi tanaman. Kemudian pengenalan hama pada tanaman ini juga dibutuhkan untuk menghindari resiko gagal panen. Beberapa teknik pengendalian dapat diterapkan untuk mengatasi serangan hama baik pengendalian secara mekanis maupun dengan penggunaan pestisida alami. Informasi pupuk dan pengenal hama beserta teknik penengalannya diperoleh berdasarkan referensi buku ”Budidaya Cabai di Lahan Sempit”[NSU-13] dan “Petunjuk Penggunaan Pupuk”[PIN-13]. Dalam mendeskripsikan langkah-langkah budidaya, pembuatan pupuk organik, dan pembuatan pestisida alami tentunya mengandung istilah-istilah dibidang pertanian yang belum diketahui. Sehingga, pengetahuan tentang istilah dibutuhkan agar dengan mudah dapat dipahami.

Pada penerapannya, proses budidaya juga membutuhkan pengetahuan tentang waktu penyiraman yang efektif. Kurang penyiraman dapat menyebabkan pertumbuhan kurang optimal yang berakibat tanaman menjadi kerdil, buah cabai menjadi kecil dan mudah gugur. Sedangkan, penyiraman secara berlebihan dapat menyebabkan pembusukan akar yang diakibatkan oleh permukaan tanah yang lembab. Prakiraan cuaca harus diperhatikan agar penyiraman secara berlebihan dapat dihindari. Sehingga, rekomendasi penyiraman dan prakiraan cuaca dibutuhkan agar pertumbuhan tanaman cabai menjadi optimal. Selain itu,

pengetahuan tentang pemupukan yang efektif juga sangat dibutuhkan. Kebutuhan hara dan nutrisi tanaman didapatkan melalui proses pemupukan. Oleh karena itu, rekomendasi pemupukan dibutuhkan untuk memaksimalkan pertumbuhan tanaman. Rekomendasi penyiraman dan pemupukan tersebut diperoleh berdasarkan referensi dari buku “Budidaya Cabai di Lahan Sempit”[NSU-13] dan “Cabe Hibrida Unggul di Lahan dan Polibag”[SAL-13] serta hasil wawancara dengan penyuluh pertanian. Berkaitan dengan proses pertumbuhan tanaman cabai, dibutuhkan kontrol terhadap waktu normal pertumbuhan agar dapat mengetahui kualitas tanaman yang baik. Estimasi pertumbuhan tersebut diperoleh berdasarkan jenis bibit yang digunakan dalam proses budidaya. Sehingga dibutuhkan pemilihan bibit yang akan digunakan dalam proses budidaya, agar estimasi waktu normal pertumbuhan dapat diperkirakan secara akurat.

Proses budidaya dapat dilakukan dari semua kalangan, termasuk seorang dengan aktifitas padat yang kemungkinan besar proses budidaya terabaikan. Oleh karena itu, pencatatan agenda kegiatan proses budidaya pada tanggal tertentu dibutuhkan sebagai pengingat. Bagi seseorang yang serius melakukan budidaya cabai secara organik untuk memperoleh keuntungan, dibutuhkan pencatatan pengeluaran selama proses budidaya untuk dilakukan analisa keuntungan pada saat panen nantinya.

4.1.2 Gambaran Umum Aplikasi

Gambaran umum aplikasi *mobile* budiaya tanaman cabai secara organik terdiri dari dua bagian, yaitu deskripsi umum aplikasi dan lingkungan aplikasi.

1. Deskripsi umum aplikasi

Aplikasi *mobile* budiaya tanaman cabai secara organik adalah aplikasi yang digunakan untuk memberikan informasi berupa langkah-langkah budidaya cabai secara organik, pembuatan pupuk organik, dan pembuatan pestisida alami. Selain itu terdapat fasilitas untuk melakukan proses budidaya dimana pengguna dapat memperoleh rekomendasi waktu penyiraman yang disesuaikan dengan cuaca dan rekomendasi waktu pemupukan. Pengguna juga dapat mencatat agenda kegiatan yang akan dilakukan pada tanggal tertentu dan mencatat pengeluaran selama proses budidaya untuk dilakukan analisa keuntungan pada saat panen nantinya.

2. Lingkungan aplikasi

Aplikasi *mobile* budidaya tanaman cabai secara organik membutuhkan suatu lingkungan yang digunakan sebagai tempat berjalannya aplikasi. Secara keseluruhan aplikasi *mobile* ini berbasis *hybrid mobile application*, sehingga membutuhkan sebuah *device* untuk menjalankan aplikasi tersebut. Dengan penggunaan teknologi *mobile smartphone* android, aplikasi *mobile* ini bisa dijalankan pada semua lingkungan *device* android.

4.1.3 Identifikasi Aktor

Pada tahap ini dilakukan identifikasi aktor yang akan berinteraksi dengan aplikasi. Pada Tabel 4.1 memperlihatkan aktor yang terlibat dan penjelasannya.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Pengguna adalah masyarakat yang dapat menggunakan aplikasi untuk melakukan budidaya tanaman cabai organik secara mandiri, melihat langkah-langkah budidaya, melihat informasi pembuatan pestisida dan pupuk organik, melakukan proses budidaya, mencatat agenda kegiatan selama budidaya, serta mencatat pengeluaran dan hasil panen.

4.1.4 Analisis Kebutuhan Fungsional

Pada daftar kebutuhan fungsional berisi spesifikasi kebutuhan fungsional pengguna yang ditunjukkan pada Tabel 4.2 dengan per-nomor menggunakan SRS (*Software Requirement Specification*).

Tabel 4.2 Spesifikasi Kebutuhan Fungsional

Nomor SRS	Kebutuhan	Fungsi
SRS_001_01	Aplikasi harus menyediakan fasilitas lihat cara budidaya, sehingga memudahkan pengguna untuk memperoleh informasi langkah-langkah budidaya secara detail.	Melihat Cara Budidaya
SRS_002_01	Aplikasi harus menyediakan fasilitas lihat hama, sehingga memudahkan pengguna untuk mengetahui gejala serangan hama terhadap tanaman cabai serta mengetahui cara pengendaliannya baik pengendali-an secara	Melihat Hama

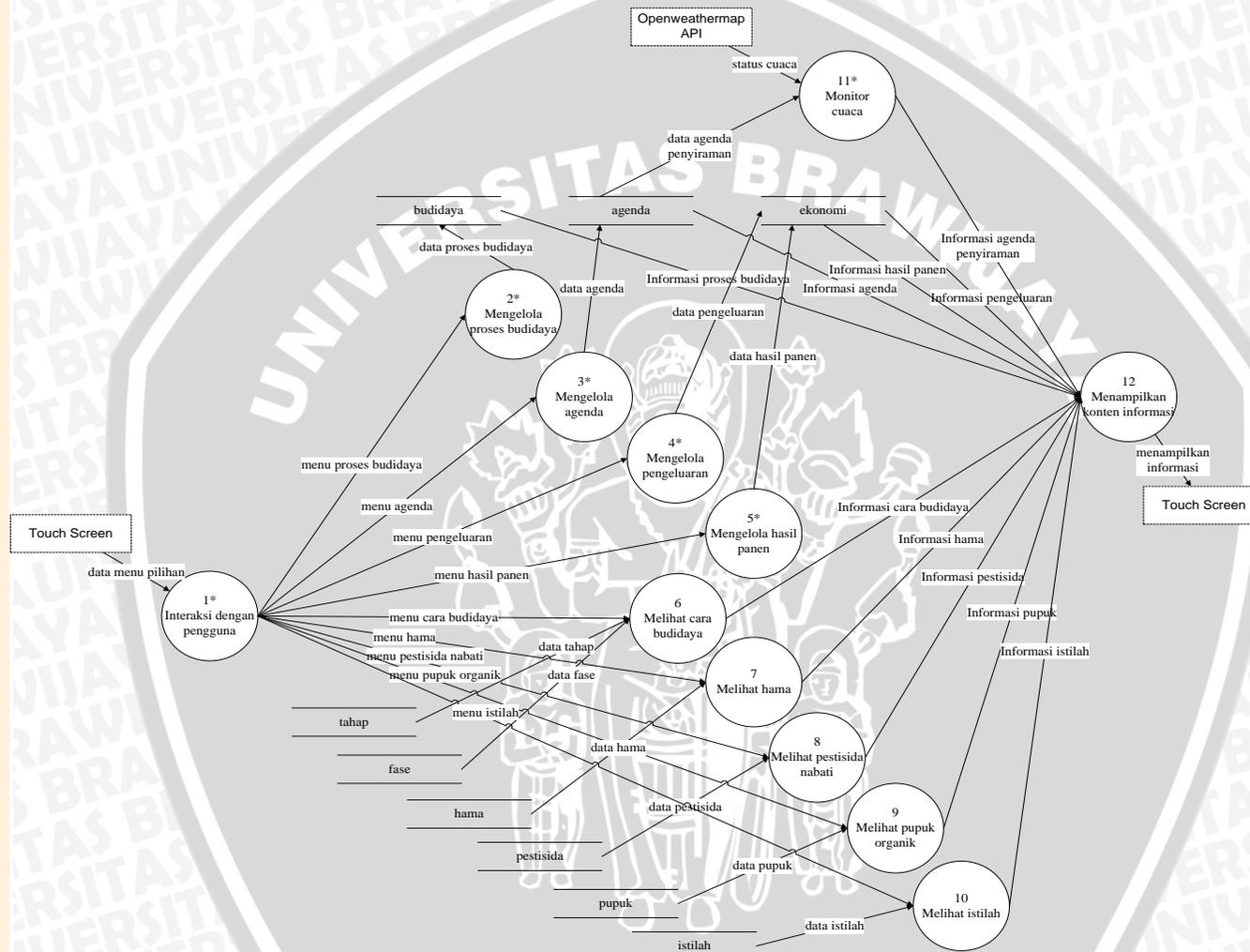
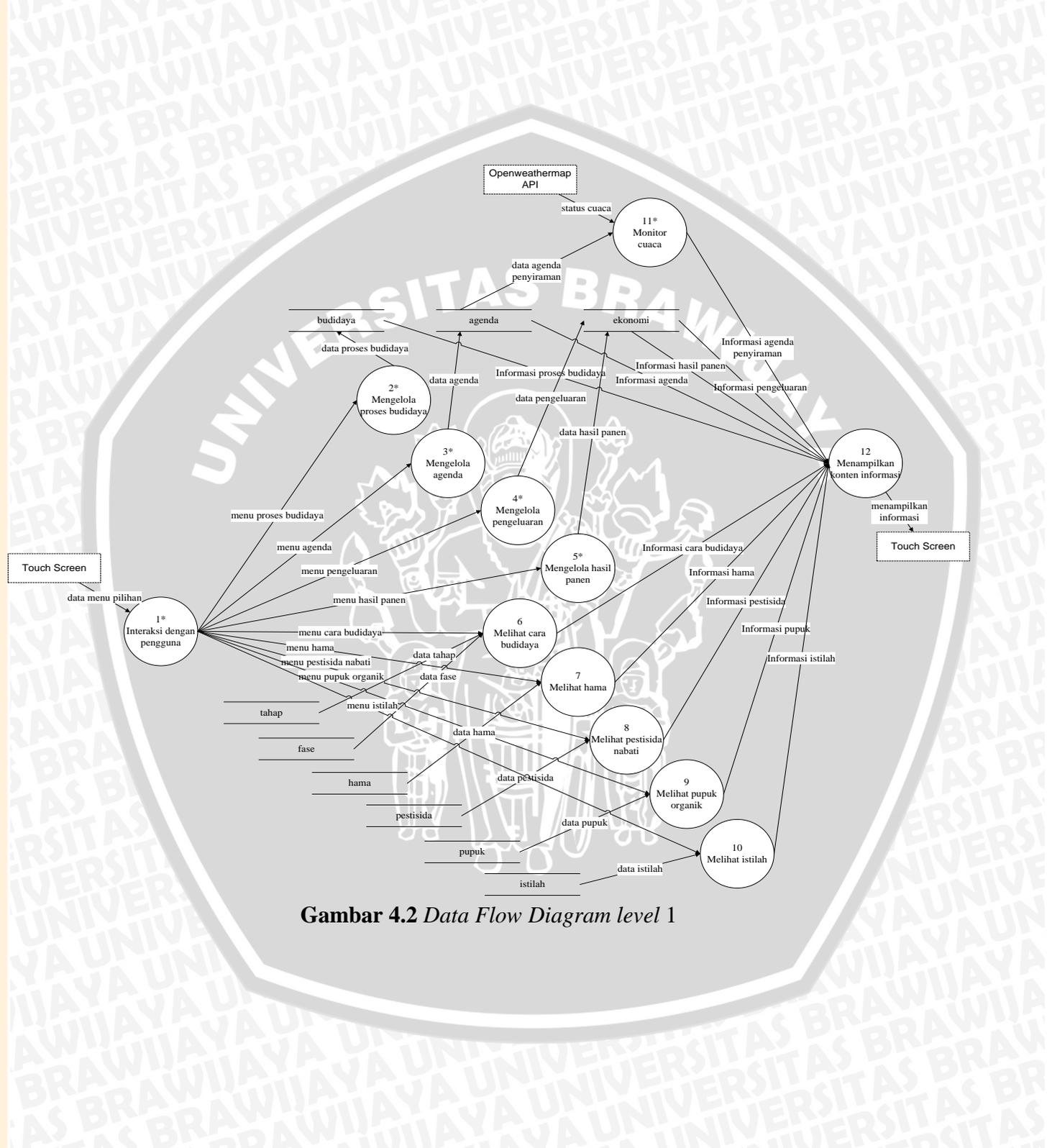
	mekanis maupun secara hayati menggunakan pestisida nabati.	
SRS_003_01	Aplikasi harus menyediakan fasilitas lihat data pestisida, sehingga memudahkan pengguna untuk mengetahui cara membuat pestisida nabati dalam mengendalikan hama yang menyerang tanaman cabai.	Melihat Pestisida Nabati
SRS_004_01	Aplikasi harus menyediakan fasilitas lihat pupuk organik, sehingga memudahkan pengguna untuk mengetahui alat, bahan, cara membuat dan cara aplikasi pupuk organik terhadap tanaman cabai.	Melihat Pupuk Organik
SRS_005_01	Aplikasi harus menyediakan fasilitas lihat data, <i>check</i> data, tambah data, dan hapus data agenda kegiatan, sehingga memudahkan pengguna dalam mengelola agenda kegiatan sebagai pengingat.	Mengelola Agenda
SRS_006_01	Aplikasi harus menyediakan fasilitas lihat data, tambah data, dan hapus data pengeluaran, sehingga memudahkan pengguna dalam mengetahui pengeluaran-pengeluaran selama proses budidaya	Mengelola Pengeluaran
SRS_007_01	Aplikasi harus menyediakan fasilitas lihat data, tambah data, dan hapus data hasil panen, sehingga memudahkan pengguna mengetahui keuntungan yang akan diperoleh saat dilakukan perhitungan terhadap total pengeluaran selama proses budidaya.	Mengelola Hasil Panen
SRS_008_01	Aplikasi harus menyediakan fasilitas mulai proses budidaya, lihat proses budidaya, dan lihat agenda harian, lihat estimasi waktu tiap fase budidaya berdasarkan benih, lihat tahap selanjutnya dan lanjut proses budidaya, serta hapus proses budidaya, sehingga me-mudahkan pengguna untuk mengetahui sejauh mana proses	Mengelola Proses Budidaya

	budidaya yang sedang dilakukan dan mengetahui beberapa informasi yang berpengaruh terhadap proses budidaya.	
SRS_008_02	Aplikasi harus menyediakan fasilitas monitor cuaca harian. Sehingga memudahkan pengguna untuk mengerjakan agenda penyiraman yang efektif.	Monitor Cuaca
SRS_009_01	Aplikasi harus menyediakan lihat istilah, sehingga memudahkan pengguna dalam memahami istilah-istilah yang digunakan dalam mendeskripsikan langkah-langkah budidaya dan pembuatan pupuk organik, maupun pembuatan pestisida nabati.	Mengelola Istilah

Selanjutnya daftar kebutuhan fungsional akan lebih dijabarkan dalam bentuk *data flow diagram* sebagai berikut.

4.1.4.1 Data flow diagram level 1

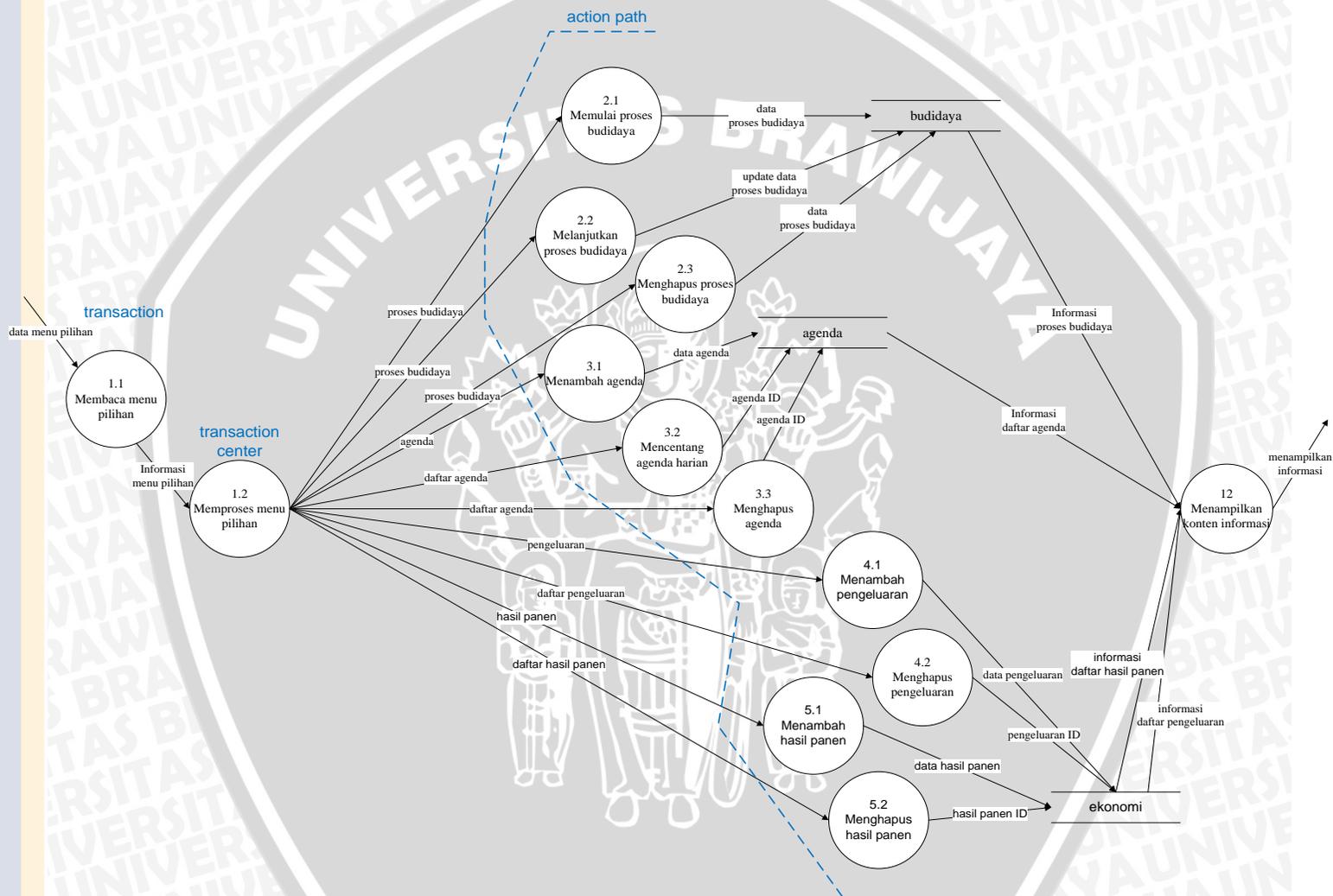
Berdasarkan *data flow diagram* pada Gambar 4.2 dijelaskan bahwa aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri memiliki 12 fungsi yaitu interaksi dengan pengguna, mengelola proses budidaya, mengelola agenda, mengelola pengeluaran, mengelola hasil panen, melihat cara budidaya, melihat hama, melihat pestisida nabati, melihat pupuk organik, melihat istilah informasi cuaca, dan menampilkan konten informasi. *Data flow diagram* tersebut memiliki karakteristik *transaction flow* dengan data menu pilihan dan status cuaca sebagai *transaction / incoming path* dan fungsi interaksi dengan pengguna sebagai *transaction center*. Sedangkan fungsi mengelola proses budidaya, mengelola agenda, mengelola pengeluaran, mengelola hasil panen, melihat cara budidaya, melihat hama, melihat pestisida nabati, melihat pupuk organik, melihat istilah, dan monitor cuaca sebagai *action path*. Penjabaran fungsi mengelola proses budidaya, mengelola agenda, mengelola pengeluaran, mengelola hasil panen, dan monitor cuaca akan dijelaskan pada *data flow diagram level 2*.



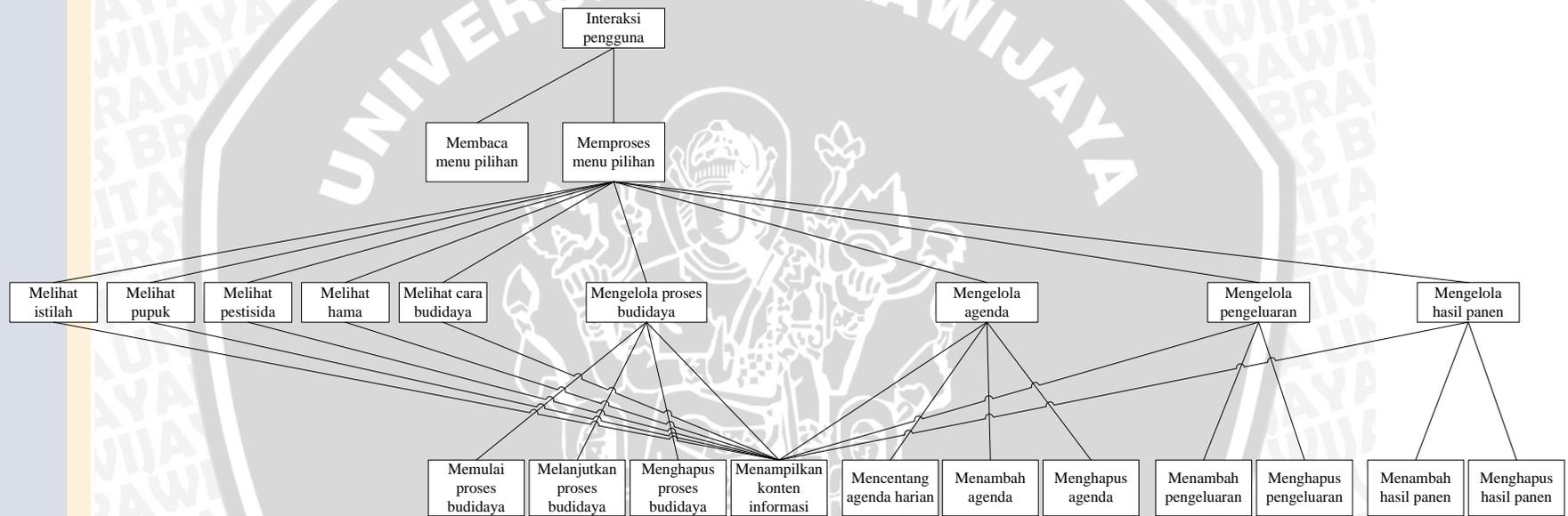
Gambar 4.2 Data Flow Diagram level 1

4.1.4.2 Data flow diagram level 2 : Interaksi dengan pengguna

Berdasarkan Gambar 4.3 dijelaskan penjabaran proses aliran data level 2 untuk fungsi interaksi dengan pengguna menjadi membaca menu pilihan dari pengguna kemudian dilakukan pemrosesan terhadap menu pilihan tersebut. Kemudian dilakukan juga penjabaran fungsi mengelola proses budidaya, mengelola agenda, mengelola pengeluaran serta mengelola hasil panen. Penjabaran fungsi mengelola proses budidaya dibagi menjadi 3 *action path* yaitu memulai proses budidaya, melanjutkan proses budidaya, dan menghapus proses budidaya dimana setiap fungsinya dilakukan pengaksesan terhadap tabel budidaya untuk menampilkan konten informasi dari proses budidaya. Selanjutnya, penjabaran fungsi mengelola agenda juga dibagi menjadi 3 *action path* yaitu menambah agenda, mencentang agenda harian, dan menghapus agenda dimana setiap fungsinya dilakukan pengaksesan terhadap tabel agenda untuk menampilkan daftar konten informasi dari agenda. Setelah itu, penjabaran fungsi mengelola pengeluaran dibagi menjadi 2 *action path* yaitu menambah pengeluaran, dan menghapus pengeluaran dimana setiap fungsinya dilakukan pengaksesan terhadap tabel ekonomi untuk menampilkan daftar konten informasi dari pengeluaran. Lalu, penjabaran fungsi mengelola hasil panen dibagi menjadi 2 *action path* yaitu menambah hasil panen, dan menghapus hasil panen dimana setiap fungsinya dilakukan pengaksesan terhadap tabel ekonomi untuk menampilkan daftar konten informasi dari hasil panen. Kemudian berdasarkan *data flow diagram level 2* tersebut, dibangun *first level factoring* untuk fungsi interaksi dengan pengguna. Data flow diagram dan *first level factoring* untuk fungsi interaksi pengguna dapat dilihat pada Gambar 4.3 dan Gambar 4.4 berikut.

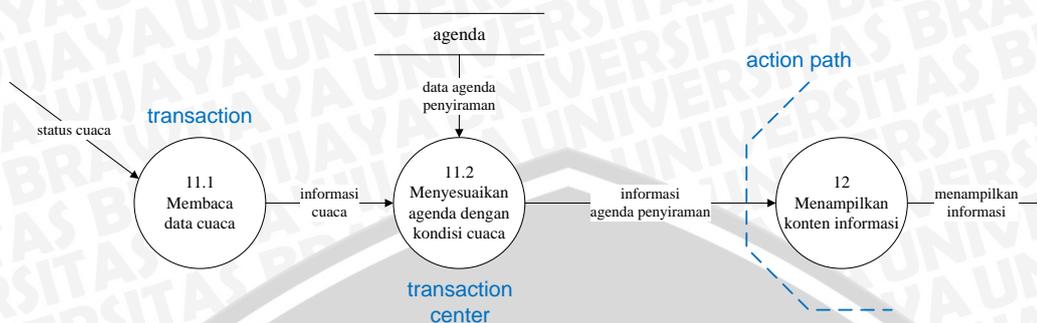


Gambar 4.3 Data flow diagram level 2 : Interaksi dengan pengguna



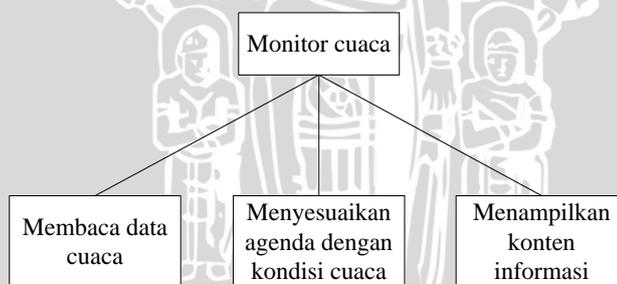
Gambar 4.4 *First level factoring* : Interaksi dengan pengguna

4.1.4.3 Data flow diagram level 2 : Monitor cuaca



Gambar 4.5 Data flow diagram level 2 : Monitor cuaca

Pada Gambar 4.5 dijelaskan proses aliran data untuk fungsi monitor cuaca yang dimulai dari membaca data cuaca berdasarkan status cuaca, kemudian sistem akan mengirimkan informasi cuaca dan pada saat yang sama dilakukan pengambilan data agenda penyiraman pada tabel agenda untuk dilakukan penyesuaian agenda penyiraman dengan kondisi cuaca. Lalu sistem akan menampilkan informasi agenda penyiraman dengan menjalankan fungsi menampilkan konten informasi. Kemudian berdasarkan *data flow diagram level 2* tersebut, dibangun *first level factoring* untuk fungsi monitor cuaca yang dapat dilihat pada Gambar 4.6 berikut.



Gambar 4.6 First level factoring : Monitor cuaca

4.1.5 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional digunakan untuk mengetahui spesifikasi yang dibutuhkan oleh sistem. Terdapat beberapa parameter dan deskripsi kebutuhan yang akan digunakan dalam pengembangan, yaitu *usability* dan *compatibility* yang akan dijelaskan pada Tabel 4.3 berikut.



Tabel 4.3 Spesifikasi Kebutuhan Non-Fungsional

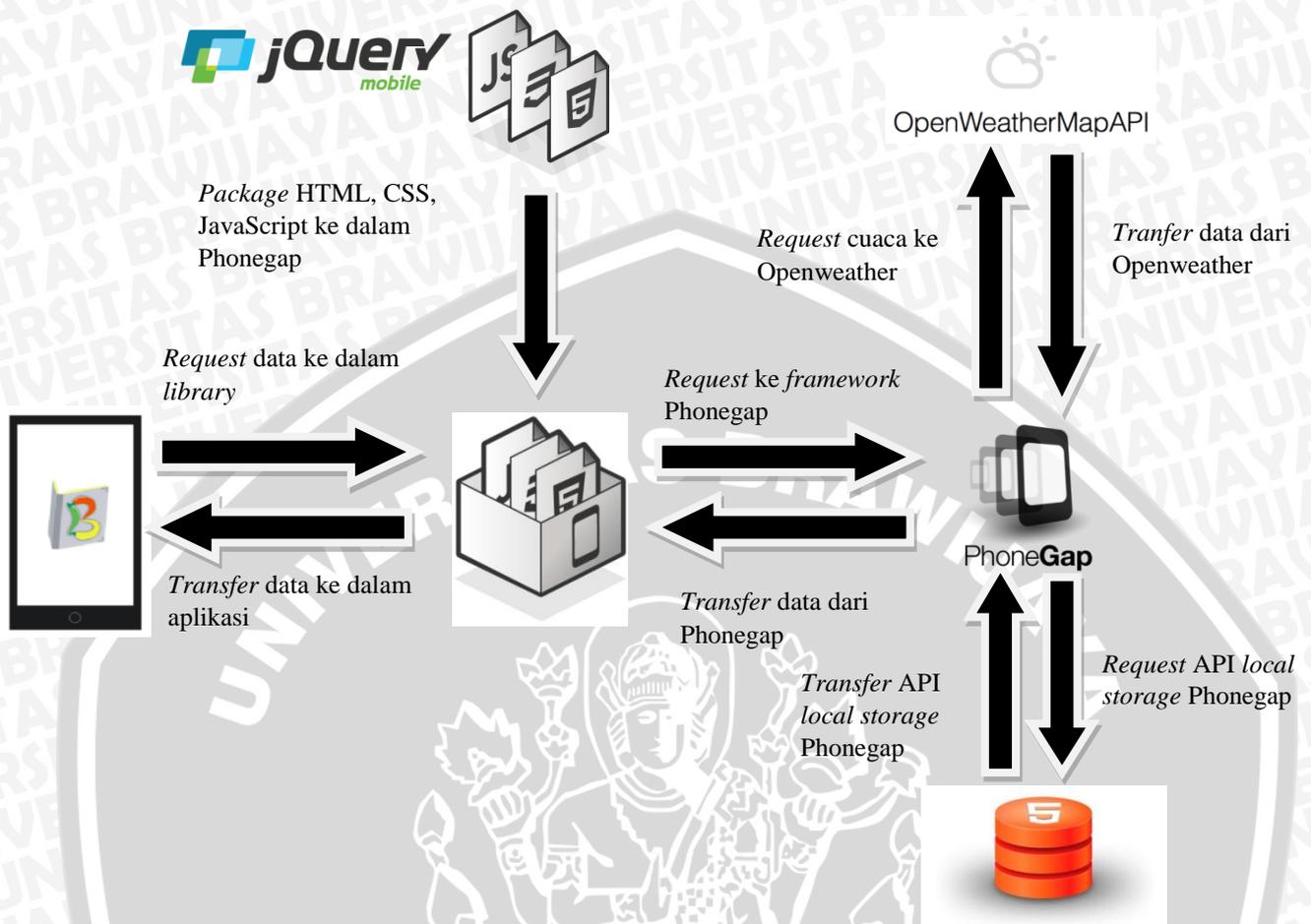
Parameter	Deskripsi Kebutuhan
<i>Compatibility</i>	Aplikasi dapat dijalankan pada berbagai versi dari sistem operasi android seperti <i>Honeycomb</i> (versi 3.0), <i>Ice Cream Sandwich</i> (versi 4.0), dan <i>Jelly Bean</i> (versi 4.3) melalui pengujian validitas fitur yang disediakan oleh sistem terkait antarmuka sistem.
<i>Usability</i>	Aplikasi dapat dengan mudah digunakan oleh pengguna melalui pengujian UAT (<i>User Acceptance Test</i>) dalam variabel pengujian yaitu kemudahan penggunaan aplikasi (<i>perceived ease of use</i>) dan pencapaian kegunaan (<i>perceived usefulness</i>). Pengujian dilakukan dengan cara pengisian kuisioner yang melibatkan pengguna dengan hasil yang diharapkan adalah persentase rata-rata penilaian setiap pernyataan dari pengguna harus lebih dari 60%

4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan dalam empat tahap, yaitu perancangan arsitektural sistem, perancangan arsitektur diagram, perancangan basis data, dan perancangan antarmuka.

4.2.1 Perancangan Arsitektur Sistem

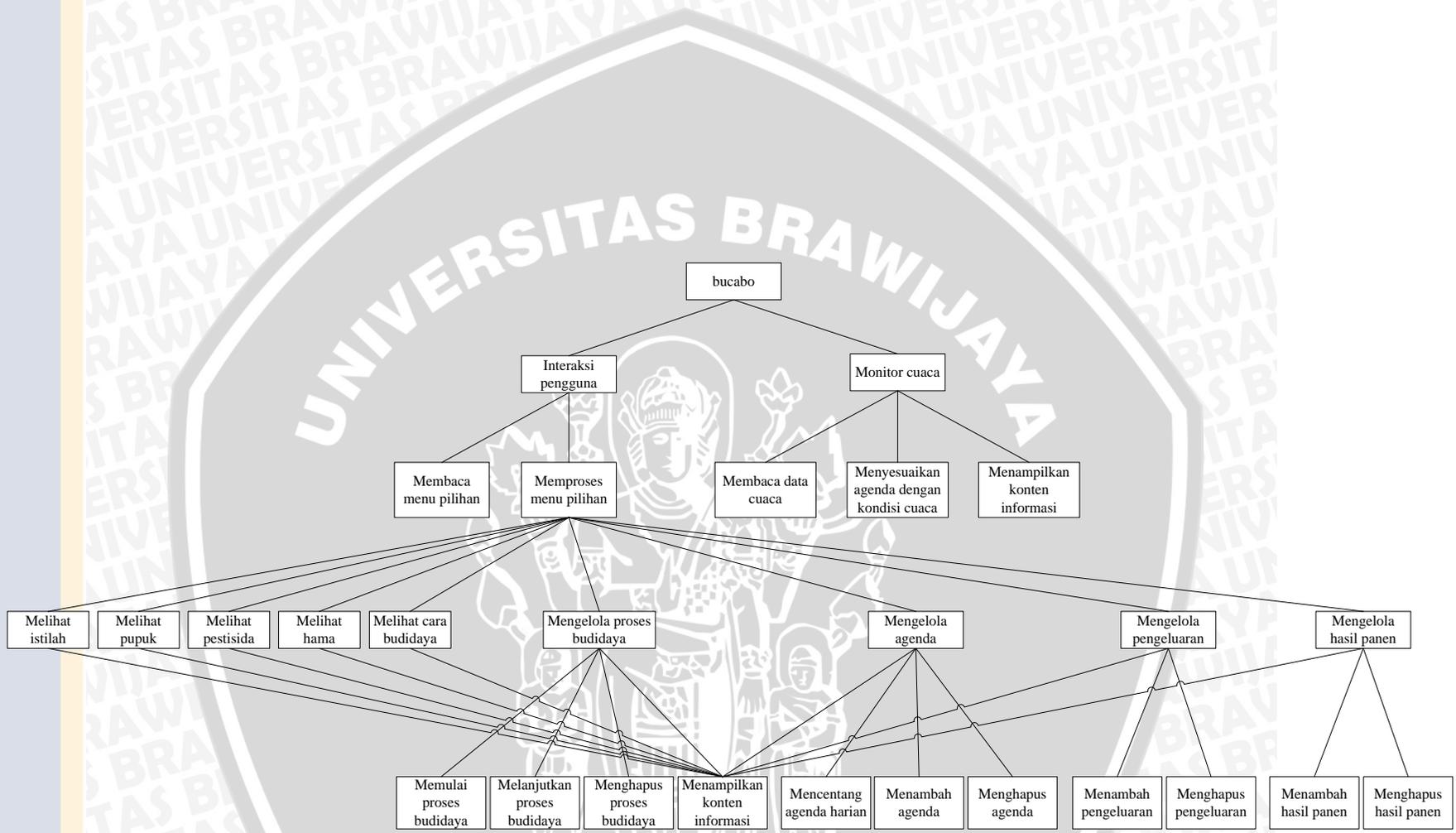
Perancangan perangkat lunak aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri dibangun dengan menggunakan konsep *hybrid mobile application* dengan pada perangkat android. Pada implementasi konsep *hybrid mobile application* ini, penulis menggunakan *framework* PhoneGap. Konsep *hybrid* dibangun menggunakan bahasa pemrograman HTML, CSS dan javascript, serta *jQuery mobile* yang memaksimalkan layout interface. Teknik penyimpanan data menggunakan *local storage* dengan DBMS SQLite. Fitur SQLite sendiri telah disediakan oleh *framework* PhoneGap dengan API *Local Storage*. Rancangan arsitektur hubungan sistem akan ditunjukkan dalam Gambar 4.7 berikut.



Gambar 4.7 Perancangan Arsitektur Sistem

4.2.2 Perancangan Arsitektur Diagram

Perancangan arsitektur diagram didasarkan oleh penggabungan dari *first level factoring* interaksi pengguna dan *first level factoring* monitor cuaca. Semua fungsi yang telah didefinisikan pada *data flow diagram level 1* dan *level 2* digambarkan dalam arsitektur diagram. Pada Gambar 4.8 terdapat 2 alur proses awal, yaitu interaksi dengan pengguna yang mengambil data input menu pilihan dari pengguna sebagai *transaction/incoming path* dan monitor cuaca dengan mengambil data status cuaca dari *web openweather* sebagai *transaction/incoming path*. Kemudian alur proses interaksi dengan pengguna dijabarkan mengikuti *first level factoring* interaksi pengguna, sedangkan alur proses monitor cuaca mengikuti *first level factoring* monitor cuaca.



Gambar 4.8 Perancangan Asitektur Diagram

4.2.3 Perancangan Basis Data

Perancangan basis data pada aplikasi ini direpresentasikan dalam bentuk *Entity Relationship Diagram* (ERD). Pada perancangan basis data sistem ini terdapat 11 buah tabel yaitu tabel budidaya, tabel agenda, tabel master fase, tabel master tahap, tabel ekonomi, tabel hama, tabel pestisida nabati, tabel pupuk organik, tabel istilah, tabel benih dan tabel cara budidaya.

Entitas fase merepresentasikan tabel *m_fase* yang berisi data fase budidaya seperti fase id dan nama fase. Struktur tabel fase ditunjukkan pada Tabel 4.4.

Tabel 4.4 Struktur Tabel Fase Budidaya

No.	Nama Field	Tipe Data	Keterangan
1	fase_id (PK)	Integer	Id dari fase budidaya
2	fase	Text	Nama dari fase budidaya

Entitas tahap merepresentasikan tabel *m_tahap* yang berisi data-data tahap budidaya seperti fase id, tahap id, nama tahap, gambar tahap, dan keterangan tahap. Struktur tabel tahap ditunjukkan pada Tabel 4.5 berikut.

Tabel 4.5 Struktur Tabel Tahap Budidaya

No.	Nama Field	Tipe Data	Keterangan
1	tahap_id (PK)	Integer	Id dari tahap budidaya
2	fase_id (FK)	Integer	Id dari fase budidaya
3	tahap	Text	Nama dari tahap budidaya
4	gambar	Text	Gambar dari tahap budidaya
5	keterangan	Text	Keterangan dari tahap budidaya

Entitas budidaya merepresentasikan tabel proses budidaya yang berisi data proses budidaya seperti fase id, tahap id, benih_id, pembaharuan terakhir, penyiraman terakhir, pemupukan terakhir dan estimasi fase. Struktur tabel proses budidaya ditunjukkan pada Tabel 4.6 berikut.

Tabel 4.6 Struktur Tabel Proses Budidaya

No.	Nama Field	Tipe Data	Keterangan
1	fase_id (FK)	Integer	Id dari fase budidaya
2	tahap_id (FK)	Integer	Id dari tahap budidaya
3	benih_id	Text	id benih yang digunakan dalam proses budidaya
3	last_update	Text	Tanggal pembaharuan terakhir dari

			proses budidaya
4	last_penyiraman	Text	Tanggal penyiraman terakhir dari proses budidaya
5	last_pemupukan	Text	Tanggal pemupukan terakhir dari proses budidaya
6	estimasi	Text	Estimasi tanggal tiap tahap dari proses budidaya

Entitas hama merepresentasikan tabel hama yang berisi data-data hama tanaman cabai seperti hama id, gejala, gambar hama, gambar gejala, jenis pestisida, deskripsi hama, dan cara pengendalian. Struktur tabel hama ditunjukkan pada Tabel 4.7 berikut.

Tabel 4.7 Struktur Tabel Hama

No.	Nama Field	Tipe Data	Keterangan
1	hama_id (PK)	Integer	Id dari hama
2	hama	Text	Nama dari hama
3	gejala	Text	Gejala dari hama
3	gambar_hama	Text	Contoh gambar dari hama
4	gambar_gejala	Text	Contoh gejala yang diakibatkan hama
5	tipe	Text	Jenis pestisida untuk pengendalian dengan pestisida nabati
6	detail	Text	Deskripsi dari hama
7	pengendalian	Text	Pengendalian secara mekanis

Entitas pestisida nabati merepresentasikan tabel pestisida nabati yang berisi data-data pestisida nabati seperti pestisida id, nama pestisida nabati, gambar tanaman, bagian tanaman, hama sasaran, keterangan jumlah bahan, dan cara pembuatan. Struktur tabel pestisida nabati ditunjukkan pada Tabel 4.8 berikut.

Tabel 4.8 Struktur Tabel Pestisida Nabati

No.	Nama Field	Tipe Data	Keterangan
1	pn_id (PK)	Integer	Id dari pestisida nabati
2	pestisida_nabati	Text	Nama dari pestisida nabati
3	gambar	Text	Gambar tanaman yang dimanfaatkan
3	bagian	Text	Bagian tanaman yang dimanfaatkan
4	sasaran	Text	Hama sasaran dari pestisida nabati

5	bahan	Text	Keterangan jumlah bahan yang digunakan
6	cara	Text	Cara pembuatan pestisida nabati

Entitas pupuk organik merepresentasikan tabel pupuk organik yang berisi data-data pupuk organik seperti nama pupuk organik, alat, bahan, cara membuat, dan cara aplikasi. Struktur tabel pupuk organik ditunjukkan pada Tabel 4.9.

Tabel 4.9 Struktur Tabel Pupuk Organik

No.	Nama Field	Type Data	Keterangan
1	nama	Text	Nama dari pupuk organik
2	alat	Text	Alat yang digunakan
3	bahan	Text	Bahan yang digunakan
3	cara_membuat	Text	Cara membuat pupuk organik
4	cara_aplikasi	Text	Cara penggunaan pupuk organik pada tanaman cabai

Entitas agenda merepresentasikan tabel agenda yang berisi data-data agenda kegiatan seperti agenda id, fase id, tahap id, tanggal agenda, nilai *ceklist* dan nama agenda. Struktur tabel agenda ditunjukkan pada Tabel 4.10 berikut.

Tabel 4.10 Struktur Tabel Agenda

No.	Nama Field	Type Data	Keterangan
1	agenda_id (PK)	Text	Id dari agenda kegiatan
2	fase_id (FK)	Integer	Id dari fase proses budidaya
3	tahap_id (FK)	Integer	Id dari tahap proses budidaya
3	agenda_tgl	Text	Tanggal dari agenda kegiatan
4	is_cek	Integer	Nilai ceklist dari agenda kegiatan
5	agenda	Text	Nama dari agenda kegiatan

Entitas ekonomi merepresentasikan tabel ekonomi yang berisi data-data pengeluaran dan hasil panen seperti ekonomi id, fase id, tahap id, jenis biaya, kategori biaya, tanggal, keterangan dan total biaya. Struktur tabel agenda ditunjukkan pada Tabel 4.11 berikut.

Tabel 4.11 Struktur Tabel Ekonomi

No.	Nama Field	Type Data	Keterangan
1	ekonomi_id (PK)	Integer	Id dari ekonomi
2	fase_id (FK)	Integer	Id dari fase proses budidaya

3	tahap_id (FK)	Integer	Id dari tahap proses budidaya
3	jenis_biaya	Text	Jenis biaya berisi jenis pembelian atau hasil panen
4	kategori_biaya	Integer	Kategori biaya berisi pengeluaran atau panen
5	tanggal	Text	Tanggal pengeluaran atau hasil panen
6	keterangan	Text	Rincian pengeluaran atau hasil panen
7	biaya	Real	Total biaya pengeluaran atau hasil panen

Entitas benih merepresentasikan tabel benih yang berisi data benih id, nama benih dan estimasi panen. Struktur tabel benih ditunjukkan pada Tabel 4.12.

Tabel 4.12 Struktur Tabel Benih

No.	Nama Field	Tipe Data	Keterangan
1	benih_id (FK)	Text	Id dari benih
2	nama	Text	Nama dari benih
3	panen	Integer	Estimasi panen setelah masa tanam

Entitas istilah merepresentasikan tabel istilah yang berisi data istilah id, nama istilah, gambar istilah, dan deskripsi istilah. Struktur tabel istilah ditunjukkan pada Tabel 4.13 berikut.

Tabel 4.13 Struktur Tabel Istilah

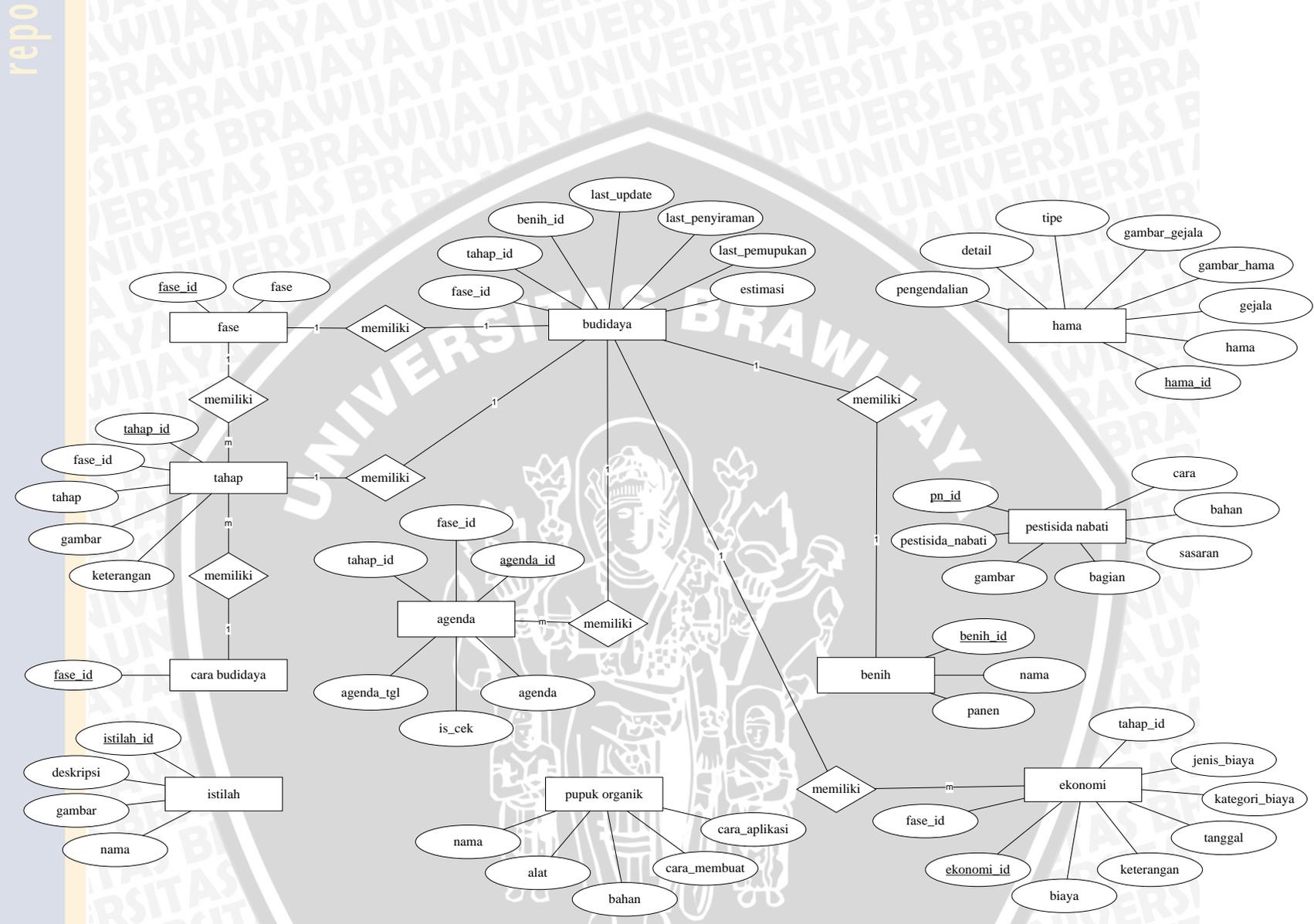
No.	Nama Field	Tipe Data	Keterangan
1	istilah_id (FK)	Integer	Id dari istilah
2	nama	Text	Nama dari istilah
3	gambar	Text	Contoh gambar dari istilah
4	deskripsi	Text	Deskripsi secara rinci dari istilah

Entitas cara budidaya merepresentasikan tabel cara budidaya yang berisi data fase id. Struktur tabel cara budidaya ditunjukkan pada Tabel 4.14 berikut.

Tabel 4.14 Struktur Tabel Cara Budidaya

No.	Nama Field	Tipe Data	Keterangan
1	fase_id (FK)	Integer	Id dari fase dari fase budidaya

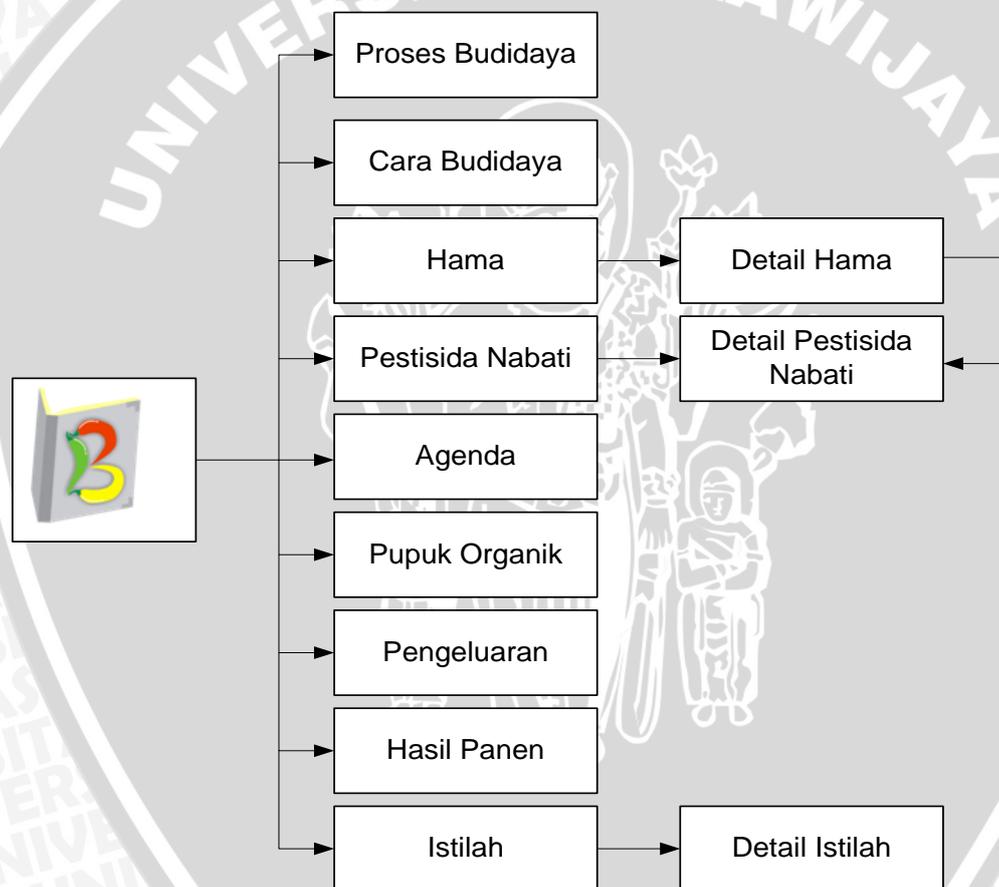
Berdasarkan penjelasan seluruh entitas yang terdapat dalam aplikasi, maka ERD dari sistem ini dapat dilihat pada Gambar 4.9 berikut.



Gambar 4.9 Entity Relationship Diagram

4.2.4 Perancangan Antarmuka

Pada bagian ini dijelaskan mengenai perancangan antarmuka aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri. Aplikasi ini akan digunakan oleh pengguna untuk mendapatkan informasi cara budidaya, pembuatan pestisida nabati, pembuatan pupuk organik, pengenalan hama, penjelasan istilah dan pencatatan agenda, pengeluaran, hasil panen serta memungkinkan untuk melakukan proses budidaya. Pada Gambar 4.10 menjelaskan *site map* dari aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri.



Gambar 4.10 Site Map Aplikasi

1. Halaman Menu Utama

Halaman menu utama merupakan halaman awal aplikasi yang terdiri dari menu-menu yang disediakan oleh aplikasi. Rancangan antarmuka halaman menu utama seperti Gambar 4.11 berikut.

1. Header	
Menu 1	
Menu 2	Menu 3
Menu 4	Menu 5
Menu 6	Menu 7
Menu 8	Menu 9
2. Daftar Menu	

Gambar 4.11 Antarmuka Halaman Menu Utama

Keterangan antarmuka halaman menu utama pada Gambar 4.11 dijelaskan pada Tabel 4.15 berikut.

Tabel 4.15 Keterangan Antarmuka Halaman Menu Utama

No.	Keterangan
1	Header aplikasi berisi logo dan nama aplikasi
2	Daftar menu, berisi menu proses budidaya, menu cara budidaya, menu istilah, menu hama, menu pestisida nabati, menu pupuk organik, menu agenda, menu pengeluaran, menu hasil panen

2. Halaman Proses Budidaya

Halaman proses budidaya merupakan halaman untuk mengelola proses budidaya yang terdiri dari informasi tahap budidaya, informasi cuaca dan informasi agenda harian tentang rekomendasi penyiraman dan pemupukan. Rancangan antarmuka halaman proses budidaya seperti Gambar 4.12 berikut.



Gambar 4.12 Antarmuka Halaman Proses Budidaya

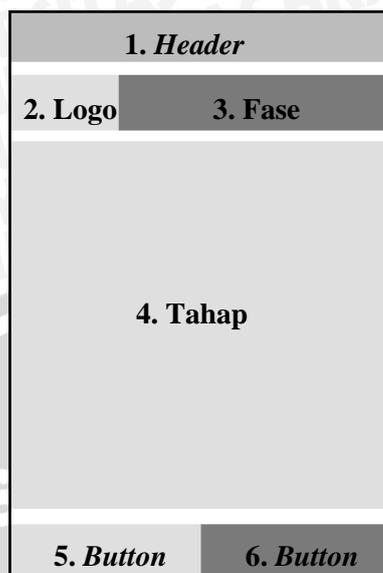
Keterangan antarmuka halaman proses budidaya pada Gambar 4.12 dijelaskan pada Tabel 4.16 berikut.

Tabel 4.16 Keterangan Antarmuka Halaman Proses Budidaya

No.	Keterangan
1	Header aplikasi berisi logo aplikasi dan <i>button</i> kembali ke halaman menu utama, sedangkan saat proses budidaya telah dimulai berisi informasi bibit yang digunakan dan estimasi waktu tiap tahap
2	Button untuk membuka panel deskripsi tentang tahap budidaya
3	Berisi nama fase dan nama tahap dari proses budidaya
4	Berisi gambar tentang tahap budidaya
5	Berisi Informasi cuaca harian
6	Berisi daftar agenda harian
7	Button hapus budidaya saat proses budidaya telah dimulai
8	Button mulai budidaya saat belum ada proses budidaya, sedangkan saat proses budidaya telah dimulai berubah menjadi button tahap selanjutnya

3. Halaman Cara Budidaya

Halaman cara budidaya merupakan halaman tutorial cara budidaya tanaman cabai yang terdiri dari informasi tahap-tahap budidaya dalam fase tertentu. Rancangan antarmuka halaman menu utama seperti Gambar 4.18 berikut.



Gambar 4.13 Antarmuka Halaman Cara Budidaya

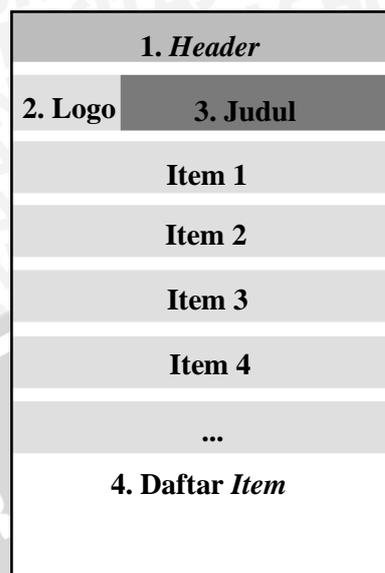
Keterangan antarmuka halaman proses budidaya pada Gambar 4.13 dijelaskan pada Tabel 4.17 berikut.

Tabel 4.17 Keterangan Antarmuka Halaman Cara Budidaya

No.	Keterangan
1	Header aplikasi berisi logo aplikasi dan <i>button</i> kembali ke halaman menu utama
2	Logo menu cara budidaya
3	Berisi nama fase dari cara budidaya
4	Berisi informasi tahap-tahap budidaya dalam fase tertentu yang terdiri dari nama tahap, gambar, dan deskripsi
5	Button fase sebelumnya
6	Button fase selanjutnya

4. Halaman Daftar

Halaman daftar merupakan halaman daftar *item* yang digunakan pada menu hama, pestisida nabati, dan istilah. Rancangan halaman daftar seperti Gambar 4.14 berikut.



Gambar 4.14 Antarmuka Halaman Daftar

Keterangan antarmuka halaman proses budidaya pada Gambar 4.14 dijelaskan pada Tabel 4.18 berikut.

Tabel 4.18 Keterangan Antarmuka Halaman Daftar

No.	Keterangan
1	Header aplikasi berisi logo aplikasi dan <i>button</i> kembali ke halaman menu utama
2	Logo menu hama, menu pestisida nabati, dan menu istilah
3	Berisi nama menu
4	Berisi daftar <i>item</i>

5. Halaman Detail

Halaman detail merupakan halaman detail yang digunakan pada menu detail-hama, menu detail-pestisida nabati, menu pupuk organik dan menu detail-istilah. Rancangan halaman detail seperti Gambar 4.15 berikut.



Gambar 4.15 Antarmuka Halaman Detail

Keterangan antarmuka halaman proses budidaya pada Gambar 4.15 dijelaskan pada Tabel 4.19 berikut.

Tabel 4.19 Keterangan Antarmuka Halaman Detail

No.	Keterangan
1	Header aplikasi berisi logo aplikasi dan button kembali ke halaman menu utama
2	Button untuk membuka panel deskripsi
3	Berisi nama dari <i>detail item</i>
4	Gambar dari <i>detail item</i>
5	Keterangan gambar <i>detail item</i>

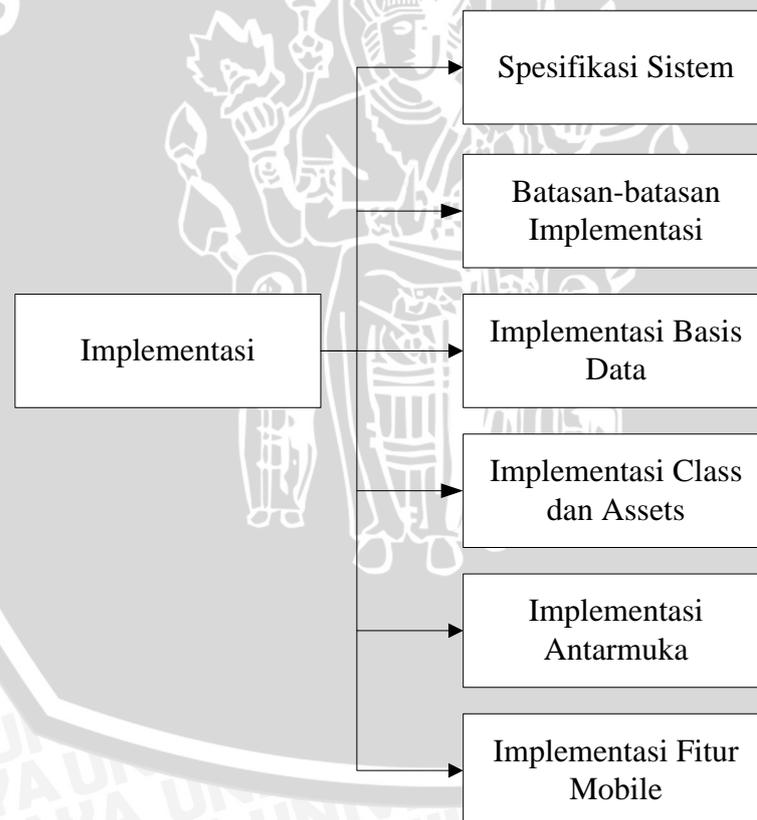
BAB V

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dibahas mengenai implementasi berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan perancangan perangkat lunak. Kemudian dilakukan pengujian dan analisis terhadap aplikasi yang telah dibangun.

5.1 Implementasi

Proses implementasi terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi *class* dan *assets* pada file program, implementasi basis data, dan implementasi antarmuka yang akan ditunjukkan dalam bentuk diagram pohon pada Gambar 5.1 berikut.



Gambar 5.1 Diagram Pohon Implementasi

5.1.1 Spesifikasi Sistem

Hasil dari analisis kebutuhan dan perancangan lunak yang telah dijelaskan pada tahap analisis kebutuhan dan tahap perancangan sistem menjadi dasar untuk dilakukan implementasi menjadi sebuah aplikasi android yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1.1 Spesifikasi Perangkat Keras

Dalam Pengembangan aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri ini menggunakan sebuah komputer dengan spesifikasi *processor*, *memory*, dan *display* yang akan dijelaskan pada Tabel 5.1 berikut.

Tabel 5.1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
<i>System Model</i>	Lenovo G405s
<i>Processor</i>	AMD A8-5550M APU @2.10Ghz (4 CPUs)
<i>Memory</i>	4096MB
<i>Display</i>	AMD Radeon HD 8550G

Dalam melakukan proses instalasi dan pengujian, perangkat yang digunakan adalah perangkat bergerak *smartphone* Android dengan spesifikasi perangkat keras yang ditunjukkan pada Tabel 5.2 berikut

Tabel 5.2 Spesifikasi Perangkat Keras *Smartphone*

Nama Komponen	Spesifikasi
<i>System Model</i>	Lenovo A316i
<i>Processor</i>	Mediatek MT6572 Dual-core 1.3 GHz Cortex-A7
<i>Memory</i>	512MB
<i>Display</i>	480 x 800 pixels, 4.0 inches (~233 ppi pixel density)

5.1.1.2 Spesifikasi Perangkat Lunak

Dalam Pengembangan aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri ini menggunakan menggunakan perangkat lunak dengan spesifikasi yang akan dijelaskan pada Tabel 5.3 berikut.

Tabel 5.3 Spesifikasi Perangkat Lunak Komputer

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 8 Pro 64-bit (6.2, Build 9200)

<i>Bahasa Pemrograman</i>	HTML, CSS dan Javascript
<i>IDE (Integrated Development Environmet)</i>	Eclipse Juno dengan ADT (Android Development Tools)

Dalam melakukan proses instalasi dan pengujian, perangkat yang digunakan adalah perangkat bergerak *smartphone* Android dengan spesifikasi perangkat lunak yang ditunjukkan pada Tabel 5.4 berikut.

Tabel 5.4 Spesifikasi Perangkat Lunak *Smartphone*

Nama Komponen	Spesifikasi
<i>Operating System</i>	Android OS, v4.2.2 (Jelly Bean)

5.1.2 Batasan-batasan implementasi

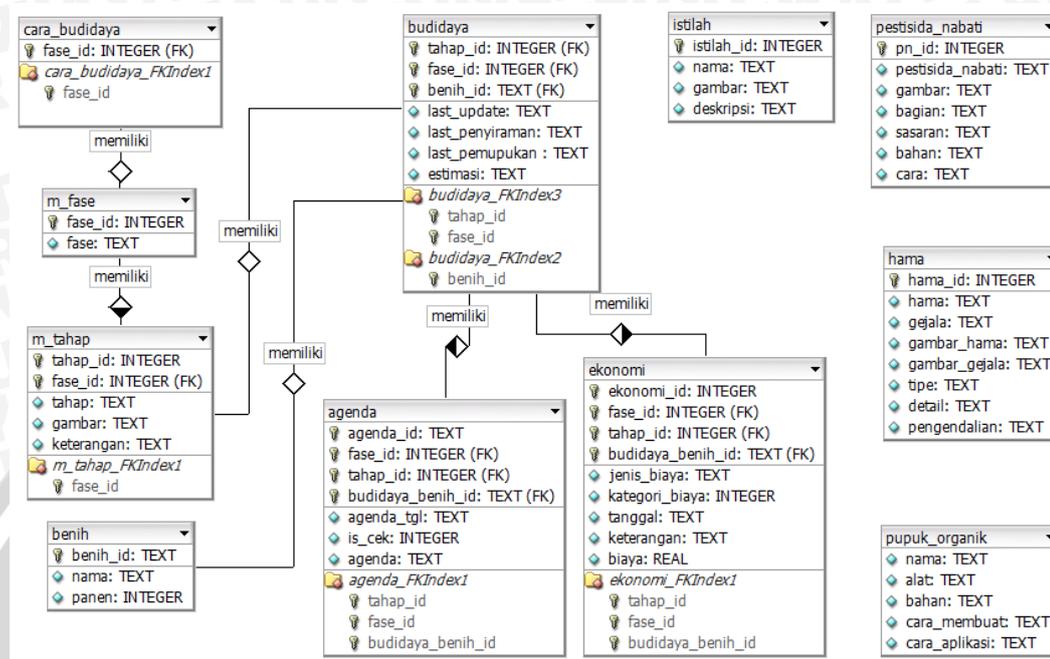
Pada implementasi perangkat lunak aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri terdapat batasan-batasan dalam proses implementasi yaitu sebagai berikut.

1. Aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri dirancang untuk dijalankan pada *smartphone* android menggunakan pendekatan pengembangan *hybrid*.
2. Dalam pembuatan aplikasi menggunakan *mobile framework* PhoneGap
3. Komunikasi data antara aplikasi dengan *local storage* diimplementasikan menggunakan pertukaran data dengan bahasa pemrograman *javascript*.
4. Penyimpanan data yang digunakan pada *local storage* adalah menggunakan SQLite pada *framework* PhoneGap.
5. Pembuatan layout pada *user interface* aplikasi menggunakan bahasa pemrograman HTML, CSS, dan Javascript.

5.1.3 Implementasi Basis Data

Implementasi pada penyimpanan data dilakukan dengan DBMS SQLite. Hasil implementasi penyimpanan data ini berupa database file dengan format *extension file* DB. Hasil implementasi SQLite pada database ini dimodelkan dalam diagram konseptual entity relationship. Pada konseptual entity relationship terdapat hubungan relasi antar tabel. Pada Gambar 5.2 menggambarkan diagram *konseptual entity relationship* dari aplikasi *mobile* panduan budidaya tanaman

cabai organik secara mandiri. Implementasi query akan ditunjukkan dalam Gambar 5.3, Gambar 5.4, dan Gambar 5.5.



Gambar 5.2 Diagram Konseptual Entity Relationship

```

1 CREATE TABLE tbl_budidaya (
2     fase_id INTEGER NOT NULL,
3     tahap_id INTEGER NOT NULL,
4     benih_id TEXT NOT NULL,
5     last_update DATE NOT NULL,
6     last_penyiraman DATE,
7     last_pemupukan DATE,
8     estimasi DATE,
9     FOREIGN KEY ("fase_id") REFERENCES "m_fase" ("fase_id"),
10    FOREIGN KEY ("tahap_id") REFERENCES "m_tahap" ("tahap_id")
11 );

```

Gambar 5.3 Implementasi Query Create tabel budidaya

```

1 CREATE TABLE tbl_agenda (
2     agenda_id TEXT NOT NULL,
3     fase_id INTEGER NOT NULL,
4     tahap_id INTEGER NOT NULL,
5     agenda_tgl DATE,
6     is_cek INTEGER NOT NULL,
7     agenda TEXT NOT NULL,

```

```

8 FOREIGN KEY ("fase_id") REFERENCES "budidaya" ("fase_id"),
9 FOREIGN KEY ("tahap_id") REFERENCES "budidaya" ("tahap_id")
10 );

```

Gambar 5.4 Implementasi Query *Create* tabel agenda

```

1 CREATE TABLE ekonomi (
2     ekonomi_id INTEGER NOT NULL,
3     fase_id INTEGER NOT NULL,
4     tahap_id INTEGER NOT NULL,
5     jenis_biaya TEXT NOT NULL,
6     kategori_biaya TEXT NOT NULL,
7     tanggal DATETIME NOT NULL,
8     keterangan TEXT,
9     biaya REAL NOT NULL,
10    FOREIGN KEY ("fase_id") REFERENCES "budidaya" ("fase_id"),
11    FOREIGN KEY ("tahap_id") REFERENCES "budidaya" ("tahap_id")
12 );

```

Gambar 5.5 Implementasi Query *Create* tabel ekonomi

5.1.4 Implementasi Class Dan Assets

Setiap class dan assets yang telah dirancang pada proses perancangan direalisasikan pada sebuah file program. Implementasi class default dengan menggunakan format java (.java) seperti pada Tabel 5.5 berikut.

Tabel 5.5 Implementasi class pada kode program *.java

Folder	Package	Nama	Nama File Program
src	example.bco	Main Activity	MainActivity.java

Pada file MainActivity.java berisi *import* bundel org.apache.cordova.* dari Phonegap agar terbaca pada sistem operasi Android. Lalu mengeset *Class Activity* ke dalam Droidgap dengan melakukan *load* url ke file index.html didalam folder *assets* agar halaman html dapat tampil sempurna pada aplikasi.

Fitur-fitur yang disediakan oleh aplikasi diantaranya lihat cara budidaya, lihat hama, lihat pestisida nabati, lihat pupuk organik, kelola proses budidaya, kelola agenda, kelola pengeluaran dan kelola hasil panen serta lihat istilah. Implementasi assets berdasarkan fitur-fitur tersebut dengan menggunakan format html (.html) seperti pada Tabel 5.6 berikut.

Tabel 5.6 Implementasi assets pada kode program *.html

Nama	Nama File Program	Keterangan
Folder Utama : assets/www/		
Cara Budidaya	cara-budidaya.html	Merupakan implementasi dari lihat cara budidaya
Proses Budidaya	beranda.html	Merupakan implementasi dari kelola proses budidaya dan informasi cuaca
Agenda	agenda-list.html	Merupakan implementasi dari kelola agenda
Pengeluaran	pemasukan.html	Merupakan implementasi dari kelola pengeluaran
Pemasukan	pengeluaran.html	Merupakan implementasi dari kelola hasil panen
Daftar Hama	hama.html	Merupakan implementasi dari lihat hama
Detail Hama	detail-hama.html	
Daftar Pestisida	pestisida.html	Merupakan implementasi dari lihat pestisida nabati
Detail Pestisida Nabati	detail-pestisida-nabati.html	
Detail Pupuk Organik	pupuk-organik.html	Merupakan implementasi dari lihat pupuk organik
Daftar Istilah	info.html	Merupakan implementasi dari lihat istilah
Detail Istilah	detail-info.html	

Kemudian implementasi assets dengan menggunakan format javascript (.js) seperti pada Tabel 5.7 berikut.

Tabel 5.7 Implementasi assets pada kode program *.js

Nama	Nama File Program	Keterangan
Folder Utama : assets/www/js/		
App	app.js	Berisi <i>query</i> pembuatan database dan <i>query</i> untuk menjalankan proses penambahan serta penghapusan data. Dan juga berisi fungsi untuk mendapatkan informasi cuaca dan lokasi. Serta menampilkan konten-konten halaman yang merupakan hasil dari <i>query</i> data ke database

Pada file app.js mengatur logika proses yang terjadi pada aplikasi dengan menjalankan baris kode yang telah didefinisikan untuk mengatur konten dan fungsi yang dibutuhkan pada tampilan tiap halaman.

5.1.5 Implementasi Antarmuka

Implementasi antarmuka aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri berdasarkan perancangan antarmuka.

1. Halaman Menu Utama

Halaman menu utama berisi daftar menu yang disediakan oleh aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri diantaranya, menu proses budidaya, menu cara budidaya, menu istilah, menu hama, menu pestisida nabati, menu pupuk organik, menu agenda, menu pengeluaran, dan menu hasil panen. Tampilan halaman menu utama seperti pada Gambar 5.6 berikut.



Gambar 5.6 Tampilan Halaman Menu Utama

2. Halaman Proses Budidaya

Halaman proses budidaya ditampilkan ketika pengguna memilih menu Proses Budidaya pada halaman menu utama. Halaman ini berisi informasi tahap budidaya, informasi cuaca, informasi benih yang digunakan dalam proses budidaya serta daftar agenda harian yang akan dikerjakan pengguna. Tampilan halaman proses budidaya seperti pada Gambar 5.7 berikut.



Gambar 5.7 Tampilan Halaman Proses Budidaya

3. Halaman Cara Budidaya

Halaman cara budidaya ditampilkan ketika pengguna memilih menu Cara Budidaya pada halaman menu utama. Halaman ini berisi informasi tahap-tahap budidaya pada fase tertentu. Tampilan halaman cara budidaya seperti pada Gambar 5.8 berikut.



Gambar 5.8 Tampilan Halaman Cara Budidaya

4. Halaman Daftar

Halaman daftar ditampilkan ketika pengguna memilih salah satu menu diantaranya menu Hama, menu Pestisida Nabati, dan menu Istilah pada halaman menu utama yang berisi daftar item dari masing-masing menu tersebut. Tampilan halaman daftar pada gambar 5.9 berikut merupakan contoh saat menu pestisida nabati dipilih.



Gambar 5.9 Tampilan Halaman Pestisida Nabati

5. Halaman Detail

Halaman detail ditampilkan ketika pengguna memilih salah satu daftar item pada halaman daftar hama, daftar pestisida nabati maupun daftar istilah. Tampilan halaman detail pada gambar 5.10 berikut merupakan contoh saat item pada menu daftar pestisida nabati dipilih.



Gambar 5.10 Tampilan Halaman Detail Pestisida Nabati

5.1.6 Implementasi Fitur *Mobile*

Fitur khusus yang dapat dijalankan pada aplikasi *mobile* yang terdapat pada penelitian ini disediakan oleh *framework* phonegap sebagai media yang dapat mengakses fitur handphone dengan pendekatan *hybrid*. Implementasi fitur khusus yang ada pada perangkat mobile pada aplikasi ini adalah fitur *geolocation* dan notifikasi cuaca.

5.1.6.1 *Geolocation*

Fitur ini akan mengakses perangkat gps pada *handphone* sehingga sistem dapat mendapatkan lokasi pengguna secara tepat dalam bentuk *longitude* dan *latitude*. Algoritma proses *geolocation* akan dijalankan bersamaan saat pengguna mengakses halaman proses budidaya. Algoritma ini disimpan dalam file javascript (*.js). Implementasi algoritma akan ditunjukkan pada kode program dalam Gambar 5.11 berikut.

```

1 function getLocation() {
2   if (navigator.geolocation) {
3     navigator.geolocation.getCurrentPosition(onSuccess,
4       onError, { enableHighAccuracy:true });
5   }

```

```
6 else{
7     alert("Perangkat anda tidak didukung fitur
geolocation");
8 }
9 }
10 function onSuccess(position) {
11     var lat=position.coords.latitude;
12     var longi=position.coords.longitude;
13     $.getJSON("http://maps.google.com/maps/api/geocode/json?
latlng="+lat+", "+longi+"&sensor=true_or_false",
14 function(data) {
15     $("#cuaca-
16 lokasi").append(data.results[3].address_components[0].long
17 _name+'<br>'+data.results[5].address_components[0].long_na
18 me);
19 });
20 }
21 function onError(error){
22     alert("Terjadi kesalahan, lokasi tidak dapat
23 diperoleh");
24 }
```

Gambar 5.11 Implementasi *Geolocation*

Gambar di atas menunjukkan implementasi dari fitur geolocationca. Pada baris ke 1 – 9 sistem melakukan pengecekan fitur *geolocation* pada *handphone*. Pada baris 10 – 20 sistem akan mengirim request lokasi ke google maps berdasarkan informasi *longitude* dan *latitude* yang telah didapatkan pada proses sebelumnya. Kemudian informasi lokasi akan ditampilkan pada halaman proses budaya seperti pada Gambar 5.12 berikut.



Gambar 5.12 Tampilan Informasi Lokasi

5.1.6.2 Notifikasi Cuaca

Fitur ini akan mengakses perangkat gps pada *handphone* sehingga sistem dapat mendapatkan lokasi pengguna secara tepat dalam bentuk *longitude* dan *latitude*. Kemudian koordinat lokasi tersebut dikirimkan ke openweathermap.org dan informasi cuaca direspon dalam format JSON. Algoritma proses notifikasi cuaca akan dijalankan bersamaan saat pengguna mengakses halaman proses budidaya. Dengan memperoleh informasi cuaca, maka rekomendasi penyiraman dapat dilakukan lebih efektif. Algoritma ini disimpan dalam file javascript (*.js). Implementasi algoritma akan ditunjukkan pada kode program dalam Gambar 5.13 berikut.

```

1  function getLocation(){
2  if (navigator.geolocation){
3    navigator.geolocation.getCurrentPosition(onSuccess,
4    onError, { enableHighAccuracy:true });
5  }
6  else{
7    alert("Perangkat anda tidak didukung fitur
geolocation");
8  }
9  }

```

```
10 function onSuccess(position) {
11     var lat=position.coords.latitude;
12     var longi=position.coords.longitude;
13     $.getJSON("http://api.openweathermap.org/data/2.5/weather?
lat="+lat+"&lon="+longi+"", function(data) {
14         var cuaca = JSON.stringify(data.weather);
15         if (cuaca==null){
16             alert('Informasi cuaca tidak dapat diperoleh');
17         }
18         else{
19             var cs = cuaca.replace('[' , ')').replace(']', '(');
20             var obj = $.parseJSON(cs);
21             var res = obj.main.toLowerCase();
22             rescuaca = res.indexOf("rain");
23             $("#cuaca-gambar").append('<br>');
24             if(res=="clouds"){
25                 $("#cuaca-keterangan").append('Berawan');}
26             else if(res=="clear"){
27                 $("#cuaca-keterangan").append('Cerah');}
28             else if(res=="rain"){
29                 $("#cuaca-keterangan").append('Hujan');
30                 $("#cuaca-hujan").append('(selesai)');
31                 $("#sr").attr("checked",true);
32                 $("#sr").attr("disabled",true);
33                 $("#sr-lbl").removeClass("ui-check-box-off");
34                 $("#sr-lbl").addClass("ui-check-box-on");
35                 $("#sr-lbl").attr("checked","checked");}
36         }
37     });
38 }
39 function onError(error){
40     alert("Terjadi kesalahan, informasi cuaca tidak dapat
41     diperoleh");
42 }
```

Gambar 5.13 Implementasi Algoritma Notifikasi Cuaca

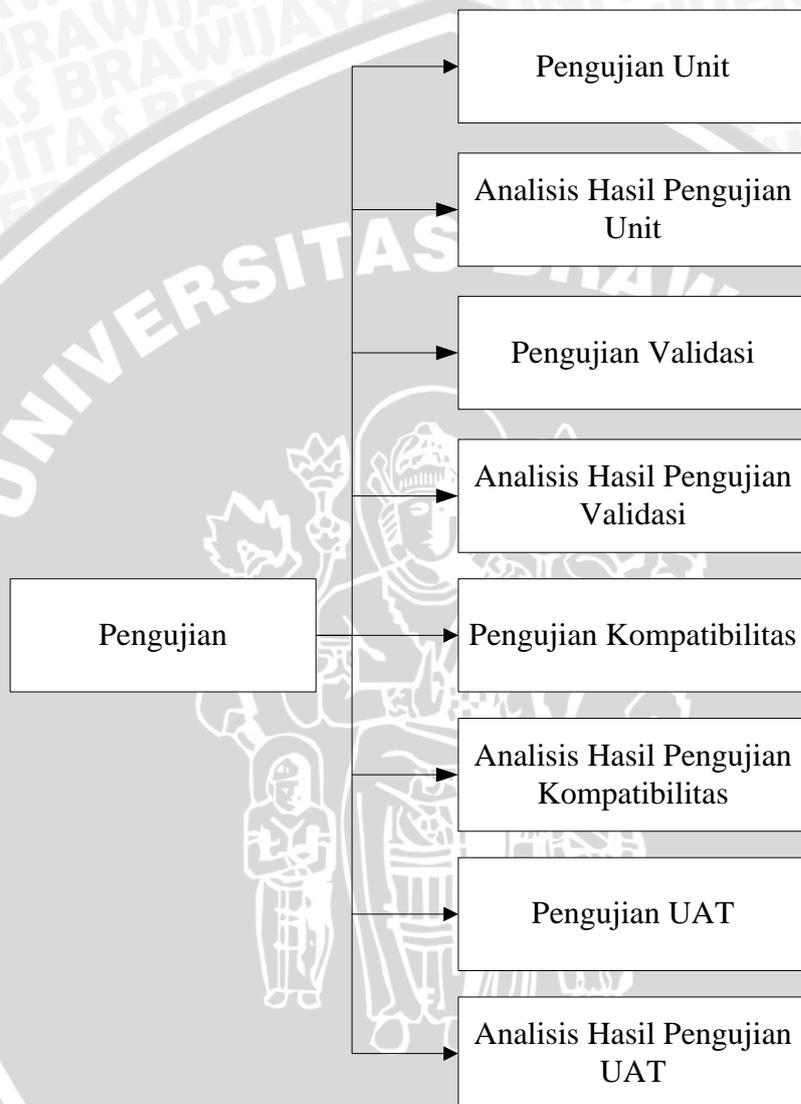
Gambar di atas menunjukkan implementasi dari fitur notifikasi cuaca. Pada baris ke 1 – 9 sistem melakukan pengecekan fitur *geolocation* pada *handphone*. Pada baris 10 – 41 sistem akan mengirim request cuaca berdasarkan informasi *longitude* dan *latitude* yang telah didapatkan pada proses sebelumnya. Kemudian informasi hasil cuaca akan ditampilkan pada halaman proses budidaya yang mempengaruhi proses penyiraman seperti pada Gambar 5.14 berikut.



Gambar 5.14 Tampilan Notifikasi Cuaca

5.2 Pengujian

Proses pengujian dilakukan melalui empat tahapan yaitu pengujian unit, pengujian validasi, pengujian kompatibilitas, dan pengujian UAT yang akan ditunjukkan dalam bentuk diagram pohon pada Gambar 5.15 berikut.



Gambar 5.15 Diagram Pohon Pengujian

5.2.1 Pengujian Unit

Pengujian unit dilakukan untuk melihat algoritma dari sistem telah berfungsi dengan benar dan sesuai. Pengujian dilakukan dengan menggunakan *tools* QUnit dengan melakukan pengujian algoritma javascript. Dalam pengujian unit ini tidak dilakukan pada semua algoritma, penulis membatasi pengujian pada algoritma

penting yang digunakan pada aplikasi yaitu fungsi monitor cuaca, menambah agenda, dan menambah hasil panen.

Kasus uji pertama adalah menguji fungsi monitor cuaca pada aplikasi. Hasil yang diharapkan dari pengujian ini adalah berjalannya fungsi monitor cuaca dengan benar dan sistem akan mendapatkan data berupa informasi cuaca. Kasus uji unit untuk fungsi monitor cuaca dapat dilihat pada Gambar 5.16 berikut.

```
1 $(document).on("pageinit", "#budidaya", function(){
2   var connection = checkConnection();
3   if (connection=='online'){
4     $("#notif-weather").append(...);
5     getLocation();
6   }
7   else if (connection=='offline'){
8     $("#notif-weather").append(...);
9   }
10  });
11  function checkConnection() {
12    var result = navigator.onLine ? 'online' : 'offline';
13    return result;
14  }
15  function getLocation(){
16    if (navigator.geolocation){
17      navigator.geolocation.getCurrentPosition(onSuccess,
18      onError, { enableHighAccuracy:true });
19    }
20    else{
21      alert("Perangkat anda tidak didukung fitur
22      geolocation");
23    }
24  }
25  function onSuccess(position) {
26    var lat=position.coords.latitude;
27    var longi=position.coords.longitude;
28    $.getJSON("http://api.openweathermap.org/data/2.5/
29    weather?lat="+lat+"&lon="+longi+"", function(data){
30      var cuaca = JSON.stringify(data.weather);
31      var cs = cuaca.replace('[','').replace(']', '');
32      var obj = $.parseJSON(cs);
33      var res = obj.main.toLowerCase();
34      rescuaca = res.indexOf("rain");
35      $("#cuaca-gambar").append('<img src=
36      "http://openweathermap.org /img/w/'+obj.icon+'.png"/>
37      <br>');
38    });
39  }
40  function onError(error){
41    alert("Terjadi kesalahan, informasi cuaca tidak dapat
42    diperoleh");
43  }
44 }
```

39	}
40	

Gambar 5.16 Kasus Uji Unit Monitor Cuaca

Pada Gambar 5.16 menjelaskan algoritma dalam bentuk kode yang akan dilakukan *testing* pada pengujian unit.

1. Pada baris 1 adalah proses inisialisasi saat halaman proses budidaya ditampilkan
2. Pada baris 2 dilakukan pengecekan koneksi dari perangkat *mobile*
3. Pada baris 3-6 adalah seleksi kondisi saat koneksi *online* dan sistem akan memanggil fungsi pengecekan lokasi
4. Pada baris 7-9 adalah seleksi kondisi saat koneksi *offline*
5. Pada baris 11-14 adalah fungsi untuk mengecek koneksi dari perangkat *mobile*
6. Pada baris 15-22 adalah fungsi untuk mendapatkan koordinat lokasi budidaya
7. Pada baris 23-36 adalah kondisi saat sistem berhasil mendapatkan koordinat lokasi, kemudian sistem akan menjalankan proses untuk mendapatkan informasi cuaca
8. Pada baris 37-39 adalah kondisi saat sistem gagal mendapatkan koordinat lokasi dan menampilkan pesan *error*

Langkah pengujian pertama adalah menggambarkan *flow graph* dari kode program pada Gambar 5.16 yang dapat dilihat pada Gambar 5.17 berikut.



Gambar 5.17 Flow Graph Monitor Cuaca

Berdasarkan *flow graph* pada Gambar 5.17, maka didapatkan nilai $V(G)$ dari *cyclomatic complexity* yaitu :

$$V(G) = \text{Jumlah Edge (20)} - \text{Jumlah Node (18)} + 2$$

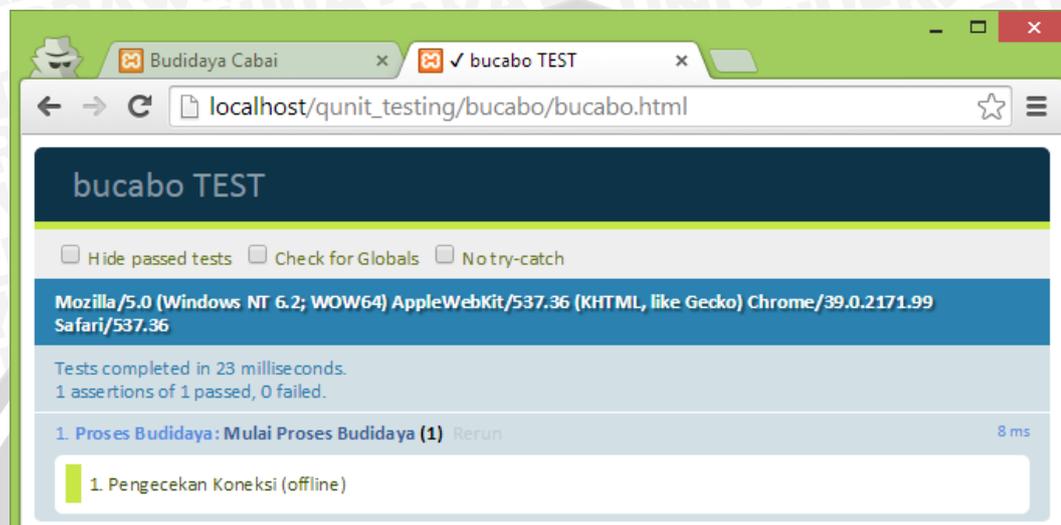
$$V(G) = 4$$

Perhitungan tersebut menjelaskan bahwa terdapat 4 *independent path* dalam *flow graph* fungsi monitor cuaca. Jadi, dibutuhkan 4 skenario pengujian untuk menguji masing-masing *path* tersebut.



Path 1 : 1-2-11-7-8-40

Skenario pengujian yang pertama dijalankan dengan keadaan koneksi perangkat *offline*. Hasil pengujian unit pada skenario pengujian pertama dijelaskan dalam Gambar 5.18 berikut.

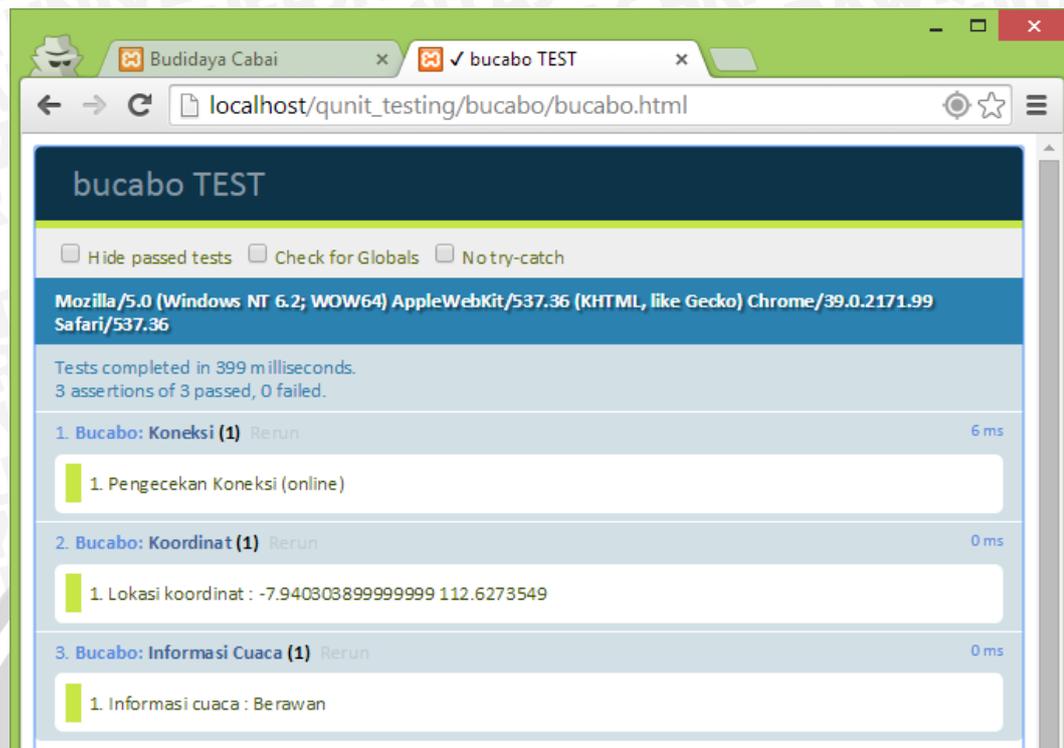


Gambar 5.18 Hasil Uji Unit Monitor Cuaca Skenario 1

Pada Gambar 5.18 menjelaskan hasil pengujian terhadap kasus uji dengan skenario yang telah disebutkan. Pengujian pada skenario pertama didapatkan alokasi waktu sebesar 23 ms dengan 1 pernyataan dinyatakan sukses karena sistem dapat menampilkan pesan offline.

Path 2 : 1-2-11-3-5-15-16-17-37-38-40

Skenario pengujian yang kedua dijalankan dengan keadaan koneksi perangkat *online*. Hasil pengujian unit pada skenario pengujian kedua dijelaskan dalam Gambar 5.19 berikut.

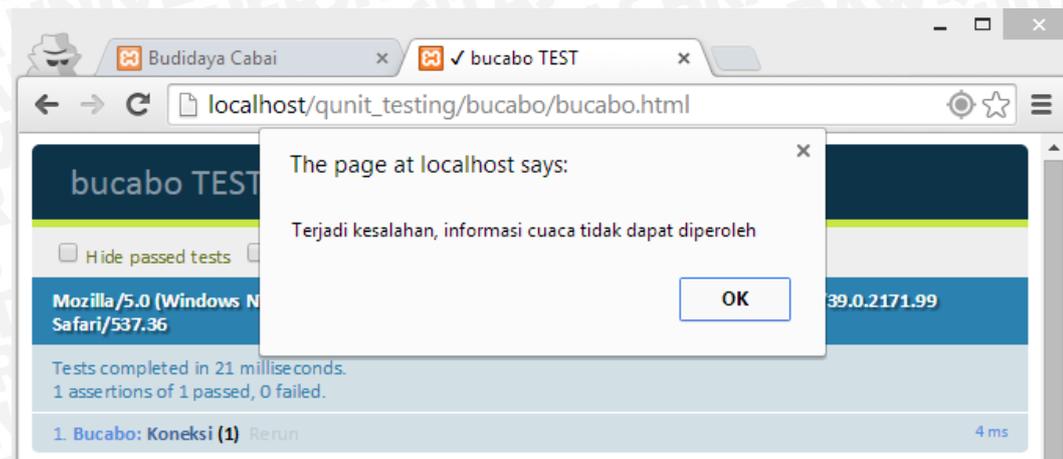


Gambar 5.19 Hasil Uji Unit Monitor Cuaca Skenario 2

Pada Gambar 5.19 menjelaskan hasil pengujian terhadap kasus uji dengan skenario yang telah disebutkan. Pengujian pada skenario kedua didapatkan alokasi waktu sebesar 399 ms dengan 3 pernyataan dinyatakan sukses dengan mendapatkan informasi cuaca.

Path 3 : 1-2-11-3-5-15-16-17-23-26-36-40

Skenario pengujian yang ketiga dijalankan dengan keadaan koneksi perangkat *online*, tetapi sistem terlalu lama mendapat informasi koordinat akibat dari koneksi yang tidak stabil. Hasil pengujian unit pada skenario pengujian ketiga dijelaskan dalam Gambar 5.20 berikut.

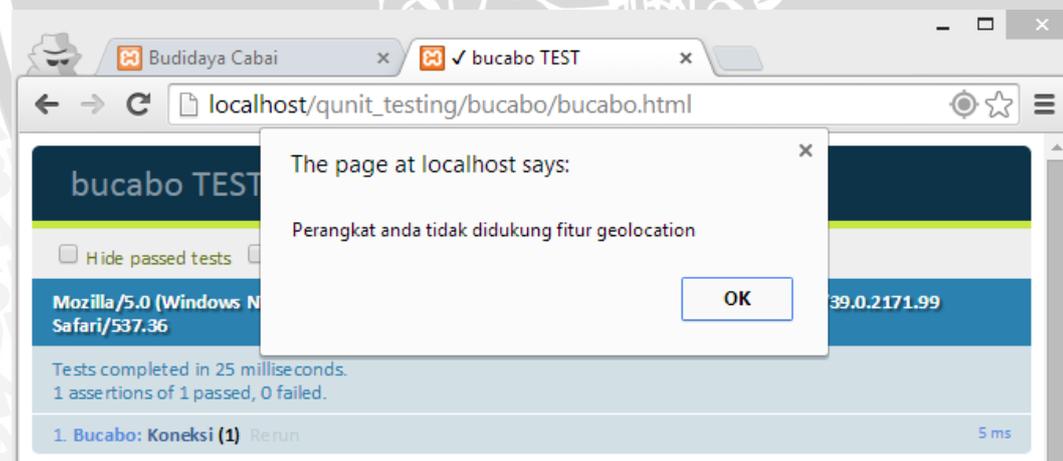


Gambar 5.20 Hasil Uji Unit Monitor Cuaca Skenario 3

Pada Gambar 5.20 menjelaskan hasil pengujian terhadap kasus uji dengan skenario yang telah disebutkan. Pengujian pada skenario ketiga didapatkan alokasi waktu sebesar 21 ms dengan 1 pernyataan dinyatakan sukses dimana sistem menampilkan pesan *error*.

Path 4 : 1-2-11-3-5-15-16-19-20-40

Skenario pengujian yang keempat dijalankan dengan perangkat yang tidak mendukung fitur geolocation. Hasil pengujian unit pada skenario pengujian keempat dijelaskan dalam Gambar 5.21 berikut.



Gambar 5.21 Hasil Uji Unit Monitor Cuaca Skenario 4

Pada Gambar 5.21 menjelaskan hasil pengujian terhadap kasus uji dengan skenario yang telah disebutkan. Pengujian pada skenario keempat didapatkan alokasi waktu sebesar 25 ms dengan 1 pernyataan dinyatakan sukses dimana sistem menampilkan pesan *error*.

Kasus uji kedua adalah menguji fungsi menambah data agenda. Hasil yang diharapkan dari pengujian ini adalah berjalannya fungsi menambah data agenda dengan benar dan sistem akan menyimpan data agenda pada *database*. Kasus uji unit untuk fungsi tambah agenda dapat dilihat pada Gambar 5.22 berikut.

```

1  var id = $("#agenda-id").val();
2  var fase = $("#fase").val();
3  var tahap = $("#tahap").val();
4  var nama = $("#nama-agenda").val();
5  var date= new Date();
6  var tanggal= date.getFullYear()+"-
  "+"0"+(date.getMonth()+1).slice(-2)+"-
  "+"0"+date.getDate().slice(-2);
7  if (nama == null || nama == "" || tanggal == null ||
  tanggal == "") {
8      alert("Silahkan isi data secara lengkap");
9      return false;
10 }
11 else{
12     db.transaction(function (tx) {
13         tx.executeSql('INSERT INTO tbl_agenda (agenda_id,
14 fase_id, tahap_id, agenda_tgl, is_cek, agenda) VALUES
15 ("'+id+'", '+fase+', '+tahap+', "'+tanggal+'", 0,
16 "'+nama+'" )');
17     });
18 }

```

Gambar 5.22 Kasus Uji Unit Tambah Agenda

Pada Gambar 5.22 menjelaskan algoritma dalam bentuk kode yang akan dilakukan testing pada pengujian unit.

1. Pada baris 1-6 adalah proses mendapatkan parameter input agenda yang akan disimpan dalam *database*
2. Pada baris 7-10 adalah seleksi kondisi saat semua inputan kosong
3. Pada baris 11-15 adalah seleksi kondisi saat inputan terisi sesuai dan query menambahkan data kedalam *database*

Langkah pengujian pertama adalah menggambarkan *flow graph* dari kode program pada Gambar 5.22 yang dapat dilihat pada Gambar 5.23 berikut.



Gambar 5.23 Flow Graph Tambah Agenda

Berdasarkan *flow graph* pada Gambar 5.23, maka didapatkan nilai $V(G)$ dari *cyclomatic complexity* yaitu :

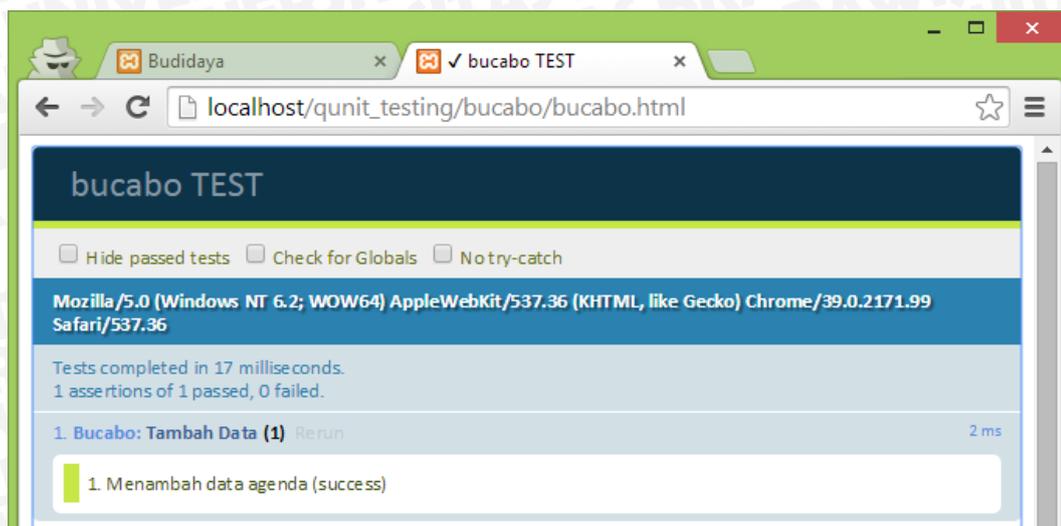
$$V(G) = \text{Jumlah Edge (13)} - \text{Jumlah Node (13)} + 2$$

$$V(G) = 2$$

Perhitungan tersebut menjelaskan bahwa terdapat 2 *independent path* dalam *flow graph* fungsi tambah agenda. Jadi, dibutuhkan 2 skenario pengujian untuk menguji masing-masing *path* tersebut.

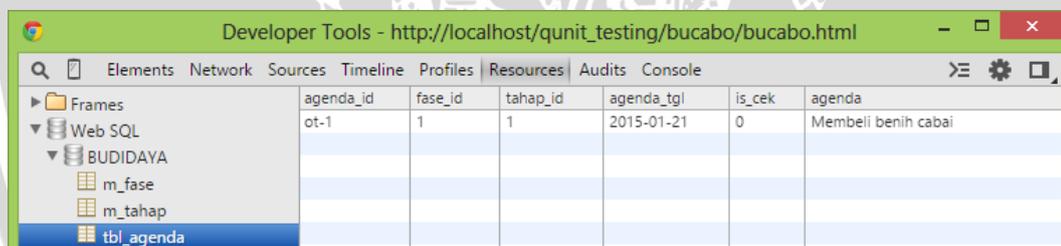
Path 1 : 1-2-3-4-5-6-7-8-9-16

Skenario pengujian yang pertama dijalankan dengan mengisi semua data inputan pada form secara benar. Hasil pengujian unit pada skenario pengujian pertama dijelaskan dalam Gambar 5.24 berikut.



Gambar 5.24 Hasil Uji Unit Tambah Agenda Skenario 1

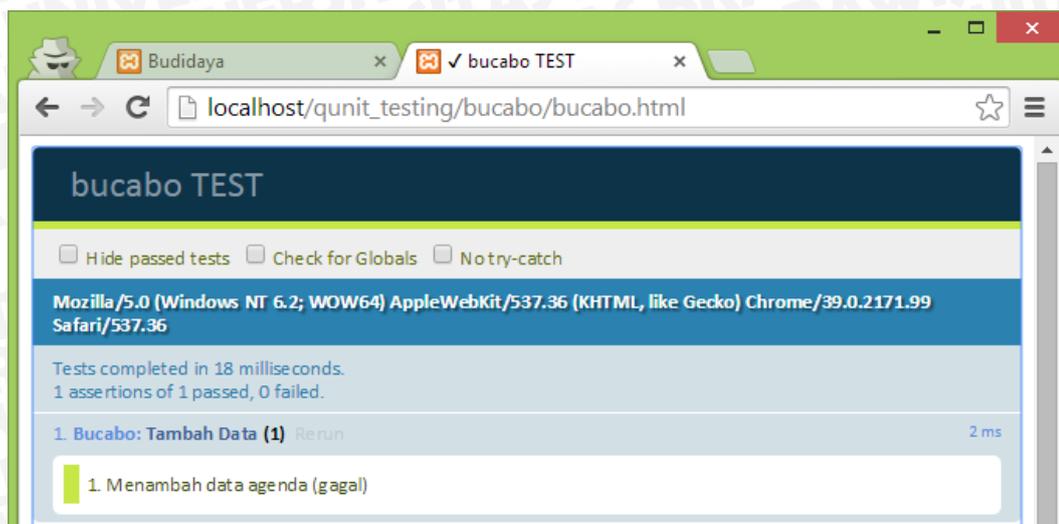
Pada Gambar 5.24 menjelaskan hasil pengujian terhadap kasus uji dengan skenario yang telah disebutkan. Pengujian pada skenario pertama didapatkan alokasi waktu sebesar 17 ms dengan 1 pernyataan dinyatakan sukses. Hasil dari *test* tersebut dapat diketahui melalui data yang masuk kedalam database yang ditunjukkan dalam Gambar 5.25 berikut.



Gambar 5.25 Hasil Uji Unit Tambah Agenda Skenario 1 pada Database

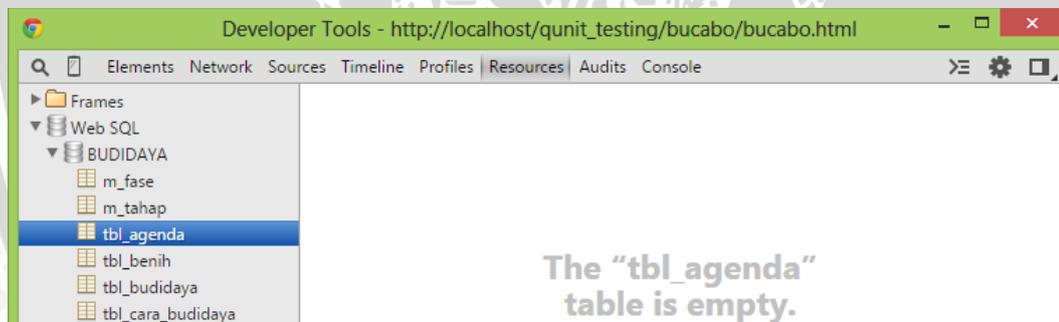
Path 2 : 1-2-3-4-5-6-7-11-12-13-16

Skenario pengujian yang kedua dijalankan dengan mengosongkan semua data inputan pada form. Hasil pengujian unit pada skenario pengujian kedua dijelaskan dalam Gambar 5.26 berikut.



Gambar 5.26 Hasil Uji Unit Tambah Agenda Skenario 2

Pada Gambar 5.26 menjelaskan hasil pengujian terhadap kasus uji dengan skenario yang telah disebutkan. Pengujian pada skenario kedua didapatkan alokasi waktu sebesar 18 ms dengan 1 pernyataan dinyatakan gagal. Hasil dari *test* tersebut dapat diketahui melalui data yang ada database tetap kosong yang ditunjukkan dalam Gambar 5.27 berikut.



Gambar 5.27 Hasil Uji Unit Tambah Agenda Skenario 2 pada Database

Kasus uji ketiga adalah menguji fungsi menambah data hasil panen. Hasil yang diharapkan dari pengujian ini adalah berjalannya fungsi menambah data hasil panen dengan benar dan sistem akan menyimpan data hasil panen pada *database*. Kasus uji unit untuk fungsi tambah hasil panen dapat dilihat pada Gambar 5.28 berikut.

```

1 var id = $("#ekonomi-id").val();
2 var fase = $("#efase").val();
3 var tahap = $("#etahap").val();
4 var hasil = $("#hasil").val();
5 var keterangan = $("#iketerangan").val();

```

```
6 var date= new Date();
7 var tanggal= date.getFullYear()+"-
  "+("0"+date.getMonth()+1).slice(-2)+"-
  "+("0"+date.getDate()).slice(-2);
8 if (hasil == null || hasil == "") {
9   alert("Silahkan isi data secara lengkap");
10  return false;
11 }
12 else if(keterangan == null || keterangan == ""){
13   keterangan = '-';
14   db.transaction(function (tx) {
15     tx.executeSql('INSERT INTO tbl_ekonomi (ekonomi_id,
fase_id, tahap_id, jenis_biaya, kategori_biaya, tanggal,
keterangan, biaya) VALUES ("'+id+'", "'+fase+'",
"'"+tahap+"", "Hasil Panen", "Panen", "'"+tanggal+"",
"'"+keterangan+"", "'"+hasil+'")');
16   })
17 }
18
```

Gambar 5.28 Kasus Uji Unit Tambah Hasil Panen

Pada Gambar 5.29 menjelaskan algoritma dalam bentuk kode yang akan dilakukan *testing* pada pengujian unit.

1. Pada baris 1-7 adalah proses mendapatkan parameter input hasil panen yang akan disimpan dalam *database*
2. Pada baris 8-11 adalah seleksi kondisi saat semua inputan kosong
3. Pada baris 12-17 adalah seleksi kondisi saat inputan keterangan kosong dan *query* menambahkan data kedalam *database*

Langkah pengujian pertama adalah menggambarkan *flow graph* dari kode program pada Gambar 5.28 yang dapat dilihat pada Gambar 5.29 berikut.



Gambar 5.29 *Flow Graph* Tambah Hasil Panen

Berdasarkan *flow graph* pada Gambar 5.29, maka didapatkan nilai $V(G)$ dari *cyclomatic complexity* yaitu :

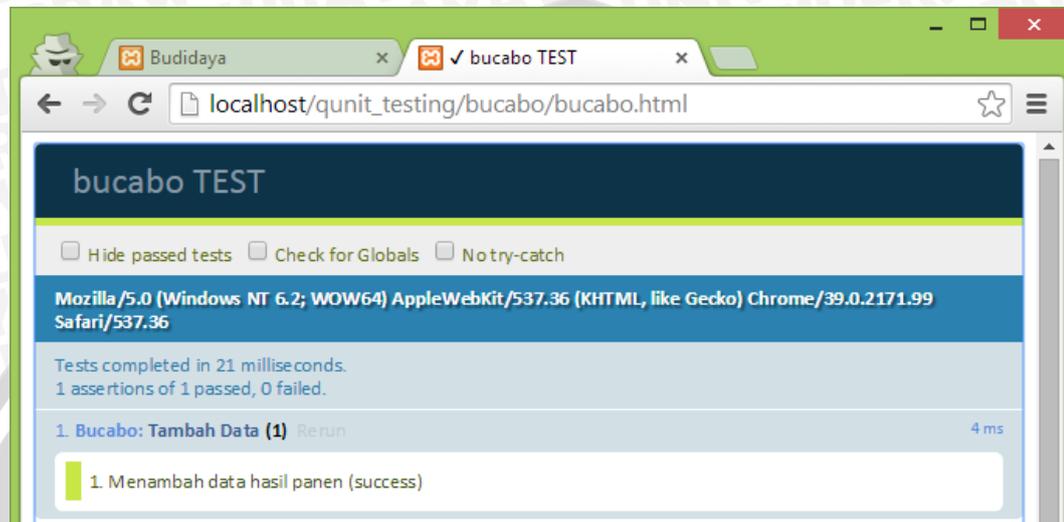
$$V(G) = \text{Jumlah Edge (14)} - \text{Jumlah Node (14)} + 2$$

$$V(G) = 14$$

Perhitungan tersebut menjelaskan bahwa terdapat 2 *independent path* dalam *flow graph* fungsi tambah hasil panen. Jadi, dibutuhkan 2 skenario pengujian untuk menguji masing-masing *path* tersebut.

Path 1 : 1-2-3-4-5-6-7-8-9-10-18

Skenario pengujian yang pertama dijalankan dengan mengisi semua data inputan pada form secara benar. Hasil pengujian unit pada skenario pengujian pertama dijelaskan dalam Gambar 5.30 berikut.



Gambar 5.30 Hasil Uji Unit Tambah Hasil Panen Skenario 1

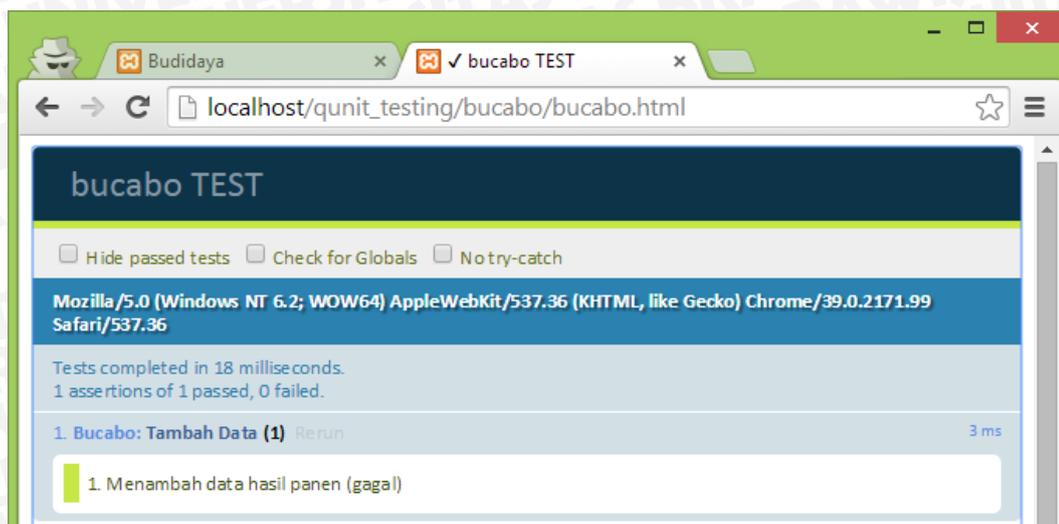
Pada Gambar 5.30 menjelaskan hasil pengujian terhadap kasus uji dengan skenario yang telah disebutkan. Pengujian pada skenario pertama didapatkan alokasi waktu sebesar 21 ms dengan 1 pernyataan dinyatakan sukses. Hasil dari *test* tersebut dapat diketahui melalui data yang masuk kedalam database yang ditunjukkan dalam Gambar 5.31 berikut.

ekonomi_id	fase_id	tahap_id	jenis_biaya	kategori_biaya	tanggal	keterangan	biaya
1	5	1	Hasil Panen	Panen	2015-01-21	5,3kg	630000

Gambar 5.31 Hasil Uji Unit Tambah Hasil Panen Skenario 1 pada *Database*

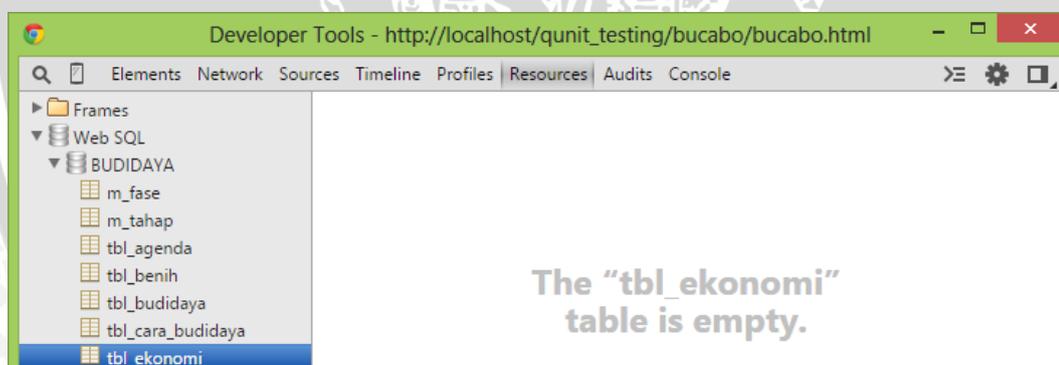
Path 2 : 1-2-3-4-5-6-7-8-12-13-15-18

Skenario pengujian yang kedua dijalankan dengan mengosongkan semua data inputan pada form. Hasil pengujian unit pada skenario pengujian kedua dijelaskan dalam Gambar 5.32 berikut.



Gambar 5.32 Hasil Uji Unit Tambah Hasil Panen Skenario 2

Pada Gambar 5.32 menjelaskan hasil pengujian terhadap kasus uji dengan skenario yang telah disebutkan. Pengujian pada skenario kedua didapatkan alokasi waktu sebesar 18 ms dengan 1 pernyataan dinyatakan gagal. Hasil dari *test* tersebut dapat diketahui melalui data yang ada database tetap kosong yang ditunjukkan dalam Gambar 5.33 berikut.



Gambar 5.33 Hasil Uji Unit Tambah Hasil Panen Skenario 2 pada Database

5.2.2 Analisis Hasil Pengujian Unit

Proses analisis terhadap hasil pengujian unit dilakukan dengan melihat kesesuaian fungsi dari implementasi unit fungsi yang diuji dengan hasil perancangan perangkat lunak yang telah dirancang sebelumnya. Terdapat 3 fungsi utama dari aplikasi yang diambil untuk diuji menggunakan *tools* QUnit, yaitu fungsi monitor cuaca, fungsi menambah agenda dan fungsi menambah pengeluaran. Tahap pengujian yang pertama adalah menggambar *flow graph* untuk mendapatkan hasil $V(G)$ dari *cyclomatic complexity* yang akan

menghasilkan kasus uji yang harus dilakukan pada satu fungsi. Pada fungsi monitor cuaca didapatkan 4 buah kasus uji dimana dari empat pengujian tersebut 4 kasus dinyatakan sukses. Pada fungsi menambah agenda terdapat 2 buah kasus uji dimana dari dua pengujian tersebut satu fungsi dinyatakan berhasil sedangkan satu fungsi lainnya dinyatakan gagal. Pada fungsi menambah hasil panen terdapat 2 buah kasus uji dimana dari dua pengujian tersebut satu fungsi dinyatakan berhasil sedangkan satu fungsi lainnya dinyatakan gagal dikarenakan sistem tidak dapat menambah data saat form tidak lengkap. Berdasarkan hal tersebut, maka dapat diambil kesimpulan bahwa unit fungsi yang diuji dari aplikasi telah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan

5.2.3 Pengujian Validasi

Pengujian validasi digunakan mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Daftar kebutuhan yang telah dirumuskan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *Black Box*, karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan. Pada pengujian validasi hanya dilakukan pengujian dengan kondisi yang normal dan tidak normal.

5.2.1.1 Kasus Uji Validasi

Pengujian validasi terhadap aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri. Untuk mengetahui kesesuaian antara daftar kebutuhan dengan kinerja sistem, pada setiap kebutuhan sistem dilakukan proses pengujian dengan kasus uji masing-masing yang akan ditunjukkan oleh tabel-tabel berikut ini.

Tabel 5.8 Kasus uji Validasi Lihat Cara Budidaya

Nomor Kasus Uji	VAL_01
Nama Kasus Uji	Kasus uji lihat cara budidaya
Objek Uji	Kebutuhan Fungsional (SRS_001_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan

	fasilitas melihat informasi langkah-langkah budidaya per tahap
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu cara budidaya
Hasil yang Diharapkan	Pengguna dapat melihat informasi langkah-langkah budidaya per tahap

Tabel 5.9 Kasus uji Validasi Lihat Hama

Nomor Kasus Uji	VAL_02
Nama Kasus Uji	Kasus uji lihat hama
Objek Uji	Kebutuhan Fungsional (SRS_002_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas melihat informasi hama secara detail
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu hama 4. Pilih salah satu daftar hama
Hasil yang Diharapkan	Pengguna dapat melihat informasi hama secara detail

Tabel 5.10 Kasus uji Validasi Lihat Pestisida Nabati

Nomor Kasus Uji	VAL_03
Nama Kasus Uji	Kasus uji lihat pestisida nabati
Objek Uji	Kebutuhan Fungsional (SRS_003_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas melihat informasi pestisida secara detail
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu pestisida nabati 4. Pilih salah satu daftar pestisida nabati
Hasil yang Diharapkan	Pengguna dapat melihat informasi pestisida nabati secara detail

Tabel 5.11 Kasus uji Validasi Lihat Pupuk Organik

Nomor Kasus Uji	VAL_04
Nama Kasus Uji	Kasus uji lihat pupuk organik
Objek Uji	Kebutuhan Fungsional (SRS_004_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas melihat informasi pupuk organik
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu pupuk organik
Hasil yang Diharapkan	Pengguna dapat melihat informasi pupuk organik

Tabel 5.12 Kasus uji Validasi Lihat Daftar Agenda

Nomor Kasus Uji	VAL_05
Nama Kasus Uji	Kasus uji lihat daftar agenda
Objek Uji	Kebutuhan Fungsional (SRS_005_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas melihat daftar agenda.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu agenda
Hasil yang Diharapkan	Pengguna dapat melihat informasi agenda

Tabel 5.13 Kasus uji Validasi Tambah Agenda dengan parameter input sesuai

Nomor Kasus Uji	VAL_06
Nama Kasus Uji	Kasus uji tambah agenda dengan semua parameter input terisi sesuai
Objek Uji	Kebutuhan Fungsional (SRS_005_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas menambah data agenda.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi

	<ol style="list-style-type: none"> 2. Masuk pada halaman menu utama 3. Pilih menu agenda 4. Masukkan data pada form dengan Nama agenda : “Membeli benih cabai” Tanggal : “2015-01-18” 5. <i>Tap button</i> Tambah
Hasil yang Diharapkan	Sistem dapat menyimpan data agenda dan ditampilkan pada daftar agenda

Tabel 5.14 Kasus uji Validasi Tambah Agenda dengan parameter input kosong

Nomor Kasus Uji	VAL_07
Nama Kasus Uji	Kasus uji tambah agenda dengan semua parameter input kosong
Objek Uji	Kebutuhan Fungsional (SRS_005_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas menambah data agenda.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu agenda 4. Masukkan data pada form dengan Nama agenda : “” Tanggal : “” 5. <i>Tap button</i> Tambah
Hasil yang Diharapkan	Sistem akan menampilkan pesan “Silahkan isi data secara lengkap”

Tabel 5.15 Kasus uji Validasi *Check* Agenda

Nomor Kasus Uji	VAL_08
Nama Kasus Uji	Kasus uji <i>check</i> agenda
Objek Uji	Kebutuhan Fungsional (SRS_005_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas melihat <i>ceklist</i> agenda yang menandakan agenda telah selesai dilakukan.

Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu agenda 4. <i>Tap</i> pada salah satu daftar agenda 5. Sistem akan menampilkan konfirmasi 6. Pilih OK
Hasil yang Diharapkan	Pengguna dapat melakukan <i>check</i> data agenda

Tabel 5.16 Kasus uji Validasi Hapus Agenda

Nomor Kasus Uji	VAL_09
Nama Kasus Uji	Kasus uji hapus agenda
Objek Uji	Kebutuhan Fungsional (SRS_005_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas menghapus data agenda tertentu.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu agenda 4. <i>Tap button</i> x pada salah satu daftar agenda 5. Sistem akan menampilkan konfirmasi penghapusan 6. Pilih OK
Hasil yang Diharapkan	Pengguna dapat menghapus data agenda

Tabel 5.17 Kasus uji Validasi Lihat Daftar Pengeluaran

Nomor Kasus Uji	VAL_10
Nama Kasus Uji	Kasus uji lihat daftar pengeluaran
Objek Uji	Kebutuhan Fungsional (SRS_006_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas melihat daftar pengeluaran.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu pengeluaran

Hasil yang Diharapkan	Pengguna dapat melihat informasi pengeluaran
------------------------------	--

Tabel 5.18 Kasus uji Validasi Tambah Pengeluaran dengan parameter input sesuai

Nomor Kasus Uji	VAL_11
Nama Kasus Uji	Kasus uji tambah pengeluaran dengan semua parameter input terisi sesuai
Objek Uji	Kebutuhan Fungsional (SRS_006_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas menambah data pengeluaran.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu pengeluaran 4. Masukkan data pada form dengan Pembelian : “Plastik Semai” Total biaya : “200000” Detail : “6 Roll” 5. <i>Tap button</i> Tambah
Hasil yang Diharapkan	Sistem dapat menyimpan data pengeluaran dan ditampilkan pada daftar pengeluaran

Tabel 5.19 Kasus uji Validasi Tambah Pengeluaran dengan parameter input kosong

Nomor Kasus Uji	VAL_12
Nama Kasus Uji	Kasus uji tambah pengeluaran dengan semua parameter input kosong
Objek Uji	Kebutuhan Fungsional (SRS_006_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas menambah data pengeluaran.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu pengeluaran 4. Masukkan data pada form dengan

	Pembelian : “” Total biaya : “” Detail : “” 5. <i>Tap button</i> Tambah
Hasil yang Diharapkan	Sistem akan menampilkan pesan “Silahkan isi data secara lengkap”

Tabel 5.20 Kasus uji Validasi Hapus Pengeluaran

Nomor Kasus Uji	VAL_13
Nama Kasus Uji	Kasus uji hapus pengeluaran
Objek Uji	Kebutuhan Fungsional (SRS_006_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas menghapus data pengeluaran tertentu.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu pengeluaran 4. <i>Tap button</i> x pada salah satu daftar pengeluaran 5. Sistem akan menampilkan konfirmasi penghapusan 6. Pilih OK
Hasil yang Diharapkan	Pengguna dapat menghapus data pengeluaran

Tabel 5.21 Kasus uji Validasi Lihat Daftar Hasil Panen

Nomor Kasus Uji	VAL_14
Nama Kasus Uji	Kasus uji lihat daftar hasil panen
Objek Uji	Kebutuhan Fungsional (SRS_007_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas melihat daftar hasil panen.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu hasil panen
Hasil yang Diharapkan	Pengguna dapat melihat informasi hasil panen

Tabel 5.22 Kasus uji Validasi Tambah Hasil Panen dengan parameter input sesuai

Nomor Kasus Uji	VAL_15
Nama Kasus Uji	Kasus uji tambah hasil panen dengan semua parameter input terisi sesuai
Objek Uji	Kebutuhan Fungsional (SRS_007_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas menambah data hasil panen.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu hasil panen 4. Masukkan informasi pada form tambah data hasil panen Hasil panen : “670000” Detail : “5,3Kg” 5. <i>Tap button</i> Tambah
Hasil yang Diharapkan	Sistem dapat menyimpan data hasil panen dan ditampilkan pada daftar hasil panen.

Tabel 5.23 Kasus uji Validasi Tambah Hasil Panen dengan parameter input kosong

Nomor Kasus Uji	VAL_16
Nama Kasus Uji	Kasus uji tambah hasil panen dengan semua parameter input kosong
Objek Uji	Kebutuhan Fungsional (SRS_007_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas menambah data hasil panen.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu hasil panen 4. Masukkan informasi pada form tambah data hasil panen Hasil panen : “” Detail : “”

	5. <i>Tap button</i> Tambah
Hasil yang Diharapkan	Sistem akan menampilkan pesan “Silahkan isi data secara lengkap”

Tabel 5.24 Kasus uji Validasi Hapus Hasil Panen

Nomor Kasus Uji	VAL_17
Nama Kasus Uji	Kasus uji hapus hasil panen
Objek Uji	Kebutuhan Fungsional (SRS_007_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas menghapus data hasil panen tertentu.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu hasil panen 4. <i>Tap button</i> x pada salah satu daftar hasil panen 5. Sistem akan menampilkan konfirmasi penghapusan 6. Pilih OK
Hasil yang Diharapkan	Pengguna dapat menghapus data hasil panen

Tabel 5.25 Kasus uji Validasi Lihat Proses Budidaya

Nomor Kasus Uji	VAL_18
Nama Kasus Uji	Kasus uji lihat proses budidaya
Objek Uji	Kebutuhan Fungsional (SRS_008_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas melihat proses budidaya.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu proses budidaya 4. <i>Tap button</i> Mulai Budidaya 5. Sistem akan menampilkan pop up berisi pemilihan bibit dan konfirmasi mulai proses budidaya 6. Pilih Mulai Budidaya
Hasil yang Diharapkan	Pengguna dapat melihat informasi proses budidaya

Tabel 5.26 Kasus uji Validasi Proses Budidaya Selanjutnya

Nomor Kasus Uji	VAL_19
Nama Kasus Uji	Kasus uji proses budidaya selanjutnya
Objek Uji	Kebutuhan Fungsional (SRS_008_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas mengubah data proses budidaya saat ini ke proses budidaya selanjutnya.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu proses budidaya 4. <i>Tap button</i> Tahap Selanjutnya 5. Sistem akan menampilkan pop up berisi konfirmasi tahap selanjutnya 6. Pilih Tahap Selanjutnya
Hasil yang Diharapkan	Pengguna dapat melanjutkan proses budidaya ke tahap selanjutnya.

Tabel 5.27 Kasus uji Validasi Hapus Hasil Panen

Nomor Kasus Uji	VAL_20
Nama Kasus Uji	Kasus uji hapus proses budidaya
Objek Uji	Kebutuhan Fungsional (SRS_008_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas menghapus data proses budidaya.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu proses budidaya 4. <i>Tap button</i> hapus 5. Sistem akan menampilkan konfirmasi penghapusan 6. Pilih Hapus Budidaya
Hasil yang Diharapkan	Pengguna dapat menghapus data proses budidaya

Tabel 5.28 Kasus uji Validasi Monitor Cuaca

Nomor Kasus Uji	VAL_21
Nama Kasus Uji	Kasus uji monitor cuaca
Objek Uji	Kebutuhan Fungsional (SRS_008_02)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas mengetahui informasi cuaca.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu proses budidaya 4. Sistem akan menampilkan informasi cuaca
Hasil yang Diharapkan	Pengguna dapat mengetahui informasi cuaca

Tabel 5.29 Kasus uji Validasi Lihat Istilah

Nomor Kasus Uji	VAL_22
Nama Kasus Uji	Kasus uji lihat istilah
Objek Uji	Kebutuhan Fungsional (SRS_009_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas melihat informasi istilah yang digunakan dalam mendeskripsikan langkah-langkah budidaya dan pembuatan pupuk organik, maupun pembuatan pestisida nabati.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka aplikasi 2. Masuk pada halaman menu utama 3. Pilih menu Istilah 4. Pilih salah satu daftar Istilah
Hasil yang Diharapkan	Pengguna dapat melihat deskripsi istilah

5.2.1.2 Hasil Pengujian Validasi

Hasil dari pengujian validasi berdasarkan 22 kasus uji yang telah didefinisikan ditunjukkan pada tabel 5.27 berikut.

Tabel 5.30 Hasil Pengujian Validasi

No.	Nomor Kasus Uji	Hasil yang Didapatkan	Status Validasi
1	VAL_01	Pengguna dapat melihat informasi langkah-langkah budidaya per tahap	VALID
2	VAL_02	Pengguna dapat melihat informasi hama secara detail	VALID
3	VAL_03	Pengguna dapat melihat informasi pestisida nabati secara detail	VALID
4	VAL_04	Pengguna dapat melihat informasi pupuk organik	VALID
5	VAL_05	Pengguna dapat melihat informasi agenda	VALID
6	VAL_06	Sistem dapat menyimpan data agenda dan ditampilkan pada daftar agenda	VALID
7	VAL_07	Sistem menampilkan pesan “Silahkan isi data secara lengkap”	VALID
8	VAL_08	Pengguna dapat melakukan <i>check</i> data agenda	VALID
9	VAL_09	Pengguna dapat menghapus data agenda	VALID
10	VAL_10	Pengguna dapat melihat informasi pengeluaran	VALID
11	VAL_11	Sistem dapat menyimpan data pengeluaran dan ditampilkan pada daftar pengeluaran	VALID
12	VAL_12	Sistem menampilkan pesan “Silahkan isi data secara lengkap”	VALID
13	VAL_13	Pengguna dapat menghapus data pengeluaran	VALID
14	VAL_14	Pengguna dapat melihat informasi hasil panen	VALID
15	VAL_15	Sistem dapat menyimpan data hasil panen dan ditampilkan pada daftar hasil panen.	VALID
16	VAL_16	Sistem menampilkan pesan “Silahkan isi data secara lengkap”	VALID
17	VAL_17	Pengguna dapat menghapus data hasil panen	VALID
18	VAL_18	Pengguna dapat melihat informasi proses budidaya	VALID
19	VAL_19	Pengguna dapat melanjutkan proses budidaya ke tahap selanjutnya.	VALID
20	VAL_20	Pengguna dapat menghapus data proses budidaya	VALID

21	VAL_21	Pengguna dapat mengetahui informasi cuaca	VALID
22	VAL_22	Pengguna dapat melihat deskripsi istilah	VALID

5.2.4 Analisis Hasil Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan melihat kesesuaian antara hasil kinerja sistem dengan daftar kebutuhan. Terdapat 19 kasus uji validasi yang dilakukan terhadap perangkat *mobile* android. Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan.

5.2.5 Pengujian Kompatibilitas

Pengujian kompatibilitas digunakan untuk mengetahui kompatibilitas antarmuka sistem pada sistem operasi dengan beberapa versi API yaitu API versi 3.0, API versi 4.0 dan API versi 4.3.

5.2.3.1 Pengujian Android Versi 3.0

Tabel 5.31 Kasus Uji dan Hasil Pengujian Kompatibilitas Android Versi 3.0

Nama Kasus Uji	Pengujian kompatibilitas Android Versi 3.0
Objek Uji	Kebutuhan Non-Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk mengetahui validitas kinerja dari sistem dalam menyediakan fitur-fitur dan antarmuka pengguna
Prosedur Uji	Membuka setiap halaman sesuai dengan spesifikasi kebutuhan sistem
Hasil yang Diharapkan	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka sistem
Hasil yang didapatkan	Sistem menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka sistem
Status Validitas	VALID

5.2.3.2 Pengujian Android Versi 4.0

Tabel 5.32 Kasus Uji dan Hasil Pengujian Kompatibilitas Android Versi 4.0

Nama Kasus Uji	Pengujian kompatibilitas Android Versi 4.0
Objek Uji	Kebutuhan Non-Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk mengetahui validitas kinerja dari

	sistem dalam menyediakan fitur-fitur dan antarmuka pengguna
Prosedur Uji	Membuka setiap halaman sesuai dengan spesifikasi kebutuhan sistem
Hasil yang Diharapkan	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka sistem
Hasil yang didapatkan	Sistem menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka sistem
Status Validitas	VALID

5.2.3.3 Pengujian Android Versi 4.3

Tabel 5.33 Kasus Uji dan Hasil Pengujian Kompatibilitas Android Versi 4.3

Nama Kasus Uji	Pengujian kompatibilitas Android Versi 4.3
Objek Uji	Kebutuhan Non-Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk mengetahui validitas kinerja dari sistem dalam menyediakan fitur-fitur dan antarmuka pengguna
Prosedur Uji	Membuka setiap halaman sesuai dengan spesifikasi kebutuhan sistem
Hasil yang Diharapkan	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka sistem
Hasil yang didapatkan	Sistem menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka sistem
Status Validitas	VALID

5.2.6 Analisis Hasil Pengujian Kompatibilitas

Proses analisis terhadap hasil pengujian kompatibilitas yang dilakukan pada aplikasi *mobile* terhadap sistem operasi Android dengan versi yang berbeda yaitu sistem operasi Android versi 3.0, versi 4.0, dan 4.2 dilakukan untuk melihat kesesuaian antara hasil kinerja sistem dengan daftar kebutuhan. Berdasarkan hasil pengujian kompatibilitas dapat disimpulkan bahwa implementasi dan fungsionalitas aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan.

5.2.7 Pengujian UAT (*User Acceptance Test*)

Pengujian UAT dilakukan dengan melibatkan pengguna yang dilakukan dengan cara pengisian kuisioner dengan jumlah pertanyaan 9 butir dan jumlah korespondensi 33 orang secara acak. Pengujian UAT bertujuan untuk melihat tingkat penerimaan pengguna terhadap aplikasi, dimana terdapat beberapa variabel penilaian yaitu kemudahan penggunaan aplikasi (*perceived ease of use*) dan pencapaian kegunaan (*perceived usefulness*). Pada Tabel 5.31 dijelaskan pernyataan-pernyataan yang berhubungan dengan variabel penilaian.

Tabel 5.34 Hasil Pengujian *Usability*

No	Pernyataan	STS	TS	N	S	SS	Total
1	Saya merasa navigasi didalam aplikasi mudah bagi saya	0	0	5	17	11	33
2	Saya merasa interaksi didalam aplikasi jelas dan mudah dimengerti	0	0	4	20	9	33
3	Saya merasa sangat mudah menggunakan aplikasi ini	0	0	5	16	12	33
4	Saya merasa aplikasi ini memberikan respon kinerja yang cepat saat dioperasikan	0	3	2	15	13	33
5	Saya merasa informasi yang ditampilkan pada aplikasi sangat jelas dan mudah dipahami	0	0	6	18	9	33
6	Aplikasi ini sangat berguna bagi saya	0	0	8	18	7	33
7	Saya merasa aplikasi ini dapat membantu saya untuk mengumpul-kan informasi secara lengkap	0	0	5	13	15	33
8	Saya akan menggunakan aplikasi ini sesering mungkin	0	1	9	13	10	33
9	Saya merasa terbantu dengan adanya aplikasi ini dalam mengetahui semua informasi kebutuhan dalam melakukan budidaya secara mandiri	0	1	5	13	14	33

Keterangan:

STS : Sangat Tidak Setuju

TS : Tidak Setuju

N : Netral

S : Setuju

SS : Sangat Setuju

5.2.8 Analisis Hasil Pengujian UAT

Proses analisis terhadap hasil pengujian UAT dilakukan dengan menggunakan skala Likert. Interpretasi skor Likert ditunjukkan pada tabel 5.32 berikut.

Tabel 5.35 Interpretasi skor Likert

Skor Likert	Interpretasi skor dengan interval = 20	Pilihan
1	0% - 19.99%	Sangat Tidak Setuju
2	20% - 39.99%	Tidak Setuju
3	40% - 59.99%	Netral
4	60% - 79.99%	Setuju
5	80% - 100%	Sangat Setuju

Keterangan:

Interval = 20 didapatkan dari pembagian nilai 100 dengan jumlah skor Likert

Kemudian dilakukan perhitungan persentase keseluruhan jawaban dari setiap pertanyaan pada quisioner dengan menggunakan perhitungan berikut.

Perhitungan Persentase Keseluruhan Jawaban (%)

$$= \frac{S_{STS} \times 1 + S_{TS} \times 2 + S_N \times 3 + S_S \times 4 + S_{SS} \times 5}{\text{Skor Likert tertinggi} \times \text{Jumlah korespondensi}} \times 100\%$$

Dengan perhitungan tersebut, didapatkan hasil berupa persentase dari setiap pertanyaan yang ditunjukkan pada Tabel 5.33 berikut.

Tabel 5.36 Persentase Keseluruhan Jawaban

No	Pernyataan	STS	TS	N	S	SS	Persentase Keseluruhan Jawaban (%)
1	Saya merasa navigasi didalam aplikasi mudah bagi saya	0	0	5	17	11	83,64
2	Saya merasa interaksi didalam aplikasi jelas dan mudah dimengerti	0	0	4	20	9	83,03
3	Saya merasa sangat mudah menggunakan aplikasi ini	0	0	5	16	12	84,24
4	Saya merasa aplikasi ini memberikan respon kinerja yang cepat saat dioperasikan	0	3	2	15	13	83,03
5	Saya merasa informasi yang	0	0	6	18	9	81,82

	ditampilkan pada aplikasi sangat jelas dan mudah dipahami						
6	Aplikasi ini sangat berguna bagi saya	0	0	8	18	7	79,39
7	Saya merasa aplikasi ini dapat membantu saya untuk mengumpulkan informasi secara lengkap	0	0	5	13	15	86,06
8	Saya akan menggunakan aplikasi ini sesering mungkin	0	1	9	13	10	79,39
9	Saya merasa terbantu dengan adanya aplikasi ini dalam mengetahui semua informasi kebutuhan dalam melakukan budidaya secara mandiri	0	1	5	13	14	84,24

Keterangan:

STS : Sangat Tidak Setuju

TS : Tidak Setuju

N : Netral

S : Setuju

SS : Sangat Setuju

Berdasarkan Tabel 5.33 pada pernyataan 1 didapatkan hasil persentase keseluruhan jawaban sebesar 83,64%. Karena sebagian besar koresponden dapat dengan memahami dengan mudah navigasi pada aplikasi, seperti semua pilihan fitur dimulai dari halaman utama. Navigasi telah diimplementasikan dengan menggunakan *icon* dan teks yang jelas. Pernyataan 2 didapatkan hasil persentase keseluruhan jawaban sebesar 83,03%. Karena interaksi pada aplikasi dibuat sederhana seperti interaksi halaman daftar hama, pestisida dan istilah ditampilkan secara jelas dalam bentuk *list* dan semua deskripsi halaman detail hama, pestisida, pupuk organik, dan istilah berada pada panel kiri. Sehingga interaksi antara halaman satu dengan yang lain tidak terlalu berbeda dan pengguna dapat dengan mudah dimengerti interaksi yang ada pada aplikasi. Pernyataan 3 didapatkan hasil persentase keseluruhan jawaban sebesar 84,24%. Karena navigasi dan interaksi pada aplikasi mudah dipahami dan digunakan oleh pengguna, sehingga aplikasi juga dapat dengan mudah digunakan oleh pengguna. Pernyataan 4 didapatkan hasil persentase keseluruhan jawaban sebesar 83,03%. Karena isi konten diakses

menggunakan *local storage* dan koneksi internet untuk mengakses cuaca menggunakan koneksi 3G sehingga aplikasi dapat memberikan respon untuk menampilkan isi konten dan fitur-fitur dengan cepat. Pernyataan 5 didapatkan hasil persentase keseluruhan jawaban sebesar 81,82%. Karena informasi yang ditampilkan terdiri dari informasi tentang cara budidaya, informasi pembuatan pupuk organik dan pestisida nabati dan informasi hama dan gejala ditampilkan secara singkat tetapi jelas, sehingga pengguna pengguna dapat memahami informasi dengan mudah. Pernyataan 6 didapatkan hasil persentase keseluruhan jawaban sebesar 79,39%. Karena aplikasi juga dapat diimplementasikan pada lahan sempit, sehingga kegunaan aplikasi tidak hanya terbatas pada lahan pertanian yang luas. Pernyataan 7 didapatkan hasil persentase keseluruhan jawaban sebesar 86,06%. Karena aplikasi menyediakan banyak informasi tentang hama dan pestisida nabati pada pengguna. Pernyataan 8 didapatkan hasil persentase keseluruhan jawaban sebesar 79,39%. Karena aplikasi memberikan rekomendasi penyiraman dan pemupukan secara rutin kepada pengguna yang sedang melakukan budidaya secara mandiri, sehingga aplikasi dapat digunakan secara intensif. Pernyataan 9 didapatkan hasil persentase keseluruhan jawaban sebesar 84,24%. Karena aplikasi memberikan panduan cara budidaya dan menyediakan informasi istilah sulit yang digunakan, memberikan rekomendasi penyiraman berdasarkan cuaca, memberikan rekomendasi pemupukan dengan menyediakan cara pembuatan pupuk, memberikan informasi hama dan menyediakan solusi penanggulangan secara mekanis maupun dengan pestisida nabati, serta menyediakan fasilitas reminder agenda, sehingga pengguna dapat terbantu dengan mendapatkan informasi tentang kebutuhan budidaya. Pengujian dinyatakan diterima ketika jumlah persentase keseluruhan jawaban lebih dari 60%. Status penerimaan pengujian ditunjukkan pada Tabel 5.34 berikut.

Tabel 5.37 Status Pengujian UAT

Aspek Penilaian	Persentase (%)	Status
Navigasi mudah	83,64%	<i>Accepted</i>
Interaksi di dalam aplikasi jelas dan mudah dimengerti	83,03%	<i>Accepted</i>
Mudah menggunakan aplikasi	84,24%	<i>Accepted</i>

Memberikan respon kinerja yang cepat	83,03%	<i>Accepted</i>
Informasi jelas dan mudah dipahami	81,82%	<i>Accepted</i>
Kegunaan aplikasi	79,39%	<i>Accepted</i>
Mendapatkan informasi yang lengkap	86,06%	<i>Accepted</i>
Penggunaan aplikasi dengan intensif	79,39%	<i>Accepted</i>
Membantu dalam mendapatkan informasi kebutuhan budidaya	84,24%	<i>Accepted</i>

Perhitungan Persentase rata-rata dari semua aspek penilaian

$$\begin{aligned} &= \frac{\text{Jumlah total persentase (\%)}}{\text{Jumlah pertanyaan}} \\ &= \frac{744,85}{9} \\ &= 82,76\% \end{aligned}$$

Kesimpulan yang didapat berdasarkan hasil perhitungan persentase keseluruhan jawaban pada setiap pertanyaan adalah aplikasi telah layak dan diterima dengan persentase rata-rata dari semua aspek penilaian sebesar 82,76%.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

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

1. Perancangan aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri telah dibuat sesuai dengan spesifikasi kebutuhan yang telah ditentukan dalam perancangan dan dapat digunakan sebagai acuan dalam melakukan budidaya cabai organik secara mandiri.
2. Sistem telah memenuhi spesifikasi kebutuhan fungsional yang ditentukan dalam perancangan berdasarkan hasil pengujian unit dan pengujian validasi dengan persentase 100% valid.
3. Sistem dinyatakan telah kompatibel dengan berbagai sistem operasi android berdasarkan hasil pengujian kompatibilitas terhadap sistem operasi pada versi 3.0, 4.0 dan 4.2.
4. Sistem telah dapat diterima oleh pengguna berdasarkan hasil pengujian UAT (*User Acceptance Testing*). Kebutuhan non-fungsional terhadap kemudahan kegunaan dan pencapaian kegunaan aplikasi dapat terpenuhi dengan persentase rata-rata dari semua aspek penilaian sebesar 82,76%.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan aplikasi *mobile* panduan budidaya tanaman cabai organik secara mandiri selanjutnya adalah :

1. Untuk pembangunan lebih lanjut, aplikasi ini dapat dikembangkan dengan menambahkan fitur *push notification* sebagai pengingat agenda.
2. Untuk pembangunan lebih lanjut, aplikasi dapat dikembangkan dengan menambahkan panduan cara budidaya dalam bentuk video agar lebih mudah dipahami.
3. Dilakukan pengembangan untuk beberapa *mobile* os lain seperti iOS dan *windows phone* agar dapat menjangkau pengguna yang menggunakan perangkat selain Android.



DAFTAR PUSTAKA

- [KOM-12] Kompas. 2012. Pasar Organik Dunia Tumbuh Pesat. WWW [terhubung berkala].
<http://health.kompas.com/read/2012/02/17/02412846/Pasar.Organik.Dunia.Tumbuh.Pesat>. [diakses pada tanggal 24 September 2014].
- [PRI-09] Priyanto. 2009. Toksikologi (Mekanisme, Terapi Antidotum dan Penilaian Resiko). Depok: Leskofi (Lembaga Studi dan Konsultasi Farmakologi).
- [ANM-10] Fakultas Pertanian dan Peternakan UIN Sultan Syarif Kasim. 2010. Pertanian Organik di Indonesia. Riau.
- [SIN-14] Sindo News. 2014. Produk Hortikultura RI Lebih Unggul di ASEAN. WWW [terhubung berkala].
<http://ekbis.sindonews.com/read/874727/34/produk-hortikultura-ri-lebih-unggul-di-asean>. [diakses tanggal 24 September 2014].
- [NSU-13] Neti Suriana. 2013, "Kandungan Nutrisi dan Manfaat Cabai bagi Kesehatan" dalam Budidaya Cabai di Lahan Sempit, eds. Raka, A., Riau, hal.32-35.
- [ASA-99] Santika, A. 1999. Agribisnis Cabai. Jakarta.
- [BCR-14] Badan Litbang Pertanian. 2014. Budidaya Tanaman Cabai Rawit. WWW [terhubung berkala].
http://hortikultura.litbang.deptan.go.id/index.php?bawaan=teknologi/isi_teknologi&id_menu=4&id_submenu=19&id=48. [diakses pada tanggal 24 September 2014].
- [BCM-14] Badan Litbang Pertanian. 2014. Budidaya Tanaman Cabai Merah. WWW [terhubung berkala].
http://hortikultura.litbang.deptan.go.id/index.php?bawaan=teknologi/isi_teknologi&id_menu=4&id_submenu=19&id=47. [diakses pada tanggal 24 September 2014].
- [ADK-02] Andoko, Agus., 2002. Budidaya Padi Secara Organik. Jakarta.
- [MUS-03] Musnamar, E. I., 2003. Pembuatan dan Aplikasi Pupuk Organik Padat. Jakarta.
- [NOV-02] Novizan. 2002. Membuat dan Memanfaatkan Pestisida Ramah Lingkungan. Jakarta: Agromedia Pustaka.

- [AIG-13] Aingindra. 2013. Pengertian Android. WWW [terhubung berkala]. <http://www.aingindra.com/android-adalah-pengertian-android-sistem-operasi.html>. [diakses pada tanggal 27 September 2014].
- [WEN-06] Willard W., 2006. *HTML: A Beginner's Guide*. McGraw-Hill.
- [SRR-08] Sulistyawan, Rubianto, Saleh R., 2008. *Modifikasi Blog Multiply Dengan CSS*. Jakarta: PT Elex Media Komputindo.
- [ANS-07] Sunyoto A., 2007. *Ajax Membangun Web dengan Teknologi Asynchronous JavaScript & XML*. Yogyakarta: Amikom Yogyakarta.
- [ZAK-12] Nicholas C. Zakas., 2012. *Professional JavaScript for Web Developers Paperback, 3rd Edition*. Wrox.
- [DAV-11] David Flanagan. 2011. *jQuery Pocket Reference Paperback, 1st Edition*. Paperback.
- [JQM-14] JQuery Mobile. 2014. WWW [terhubung berkala]. <http://jquerymobile.com/>. [diakses pada tanggal 27 September 2014].
- [DIK-07] Setiawan D., 2007. *Mengenal SQLite Command Line*. Malang: Universitas Islam Negeri Maulana Malik Ibrahim
- [DEV-11] Dev Xbata., 2011. *Mengenal database SQLite*. WWW [terhubung berkala]. <http://dev.xbata.com/database/mengenal-database-sqlite>. [diakses pada tanggal 27 September 2014].
- [PHG-14] PhoneGap. 2014. WWW [terhubung berkala]. <http://docs.phonegap.com/en/4.0.0/index.html>. [diakses pada tanggal 27 September 2014].
- [WRG-12] John M. Wargo., 2012. *PhoneGap Essentials: Building Cross-platform Mobile App*. Addison-Wesley Professional
- [PIN-13] Lingga, Pinus., 2013. *Petunjuk Penggunaan Pupuk*. Jakarta.
- [SAL-13] Salim, Emil., 2013. *Bertanam Cabe Hibrida Unggul di Lahan dan Polibag*. Yogyakarta
- [SOM-07] Sommerville, Ian., 2007. *Software Engineering 8th Edition*.

LAMPIRAN

Lampiran 1. Lembar Kuisioner UAT

QUESTIONER PENGUJIAN APLIKASI

BUCABO (Budidaya Cabai Organik)

Nama :

.....

Jenis Kelamin : L / P

Umur :

Berilah tanda silang (x) pada jawaban yang anda anggap paling sesuai!

No	Pernyataan	STS	TS	N	S	SS
1	Saya merasa navigasi didalam aplikasi mudah bagi saya					
2	Saya merasa interaksi didalam aplikasi jelas dan mudah dimengerti					
3	Saya merasa sangat mudah menggunakan aplikasi ini					
4	Saya merasa aplikasi ini memberikan respon kinerja yang cepat saat dioperasikan					
5	Saya merasa informasi yang ditampilkan pada aplikasi sangat jelas dan mudah dipahami					
6	Aplikasi ini sangat berguna bagi saya					
7	Saya merasa aplikasi ini dapat membantu saya untuk mengumpulkan informasi secara lengkap					
8	Saya akan menggunakan aplikasi ini sesering mungkin					
9	Saya merasa terbantu dengan adanya aplikasi ini dalam mengetahui semua informasi kebutuhan dalam melakukan budidaya secara mandiri					

Keterangan:

STS : Sangat Tidak Setuju TS : Tidak Setuju N : Netral

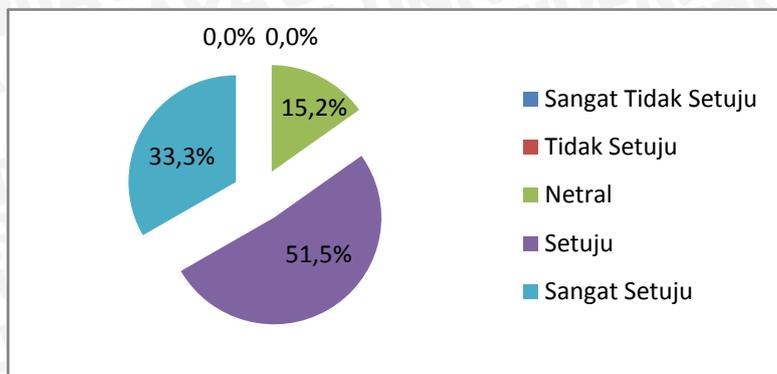
S : Setuju SS : Sangat Setuju

Malang,

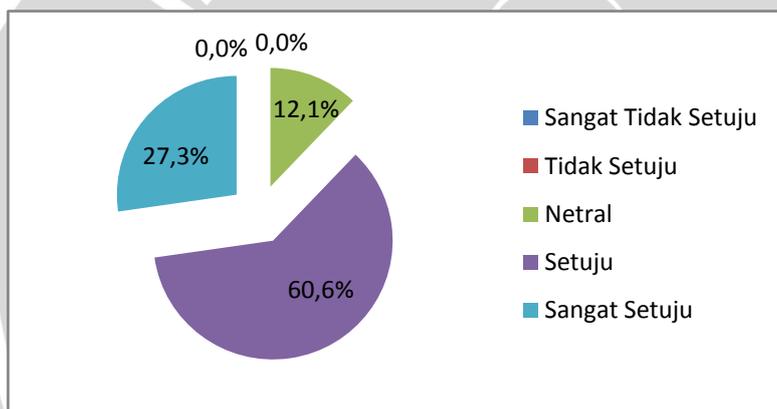
.....

Lampiran 2. Hasil Pengujian UAT

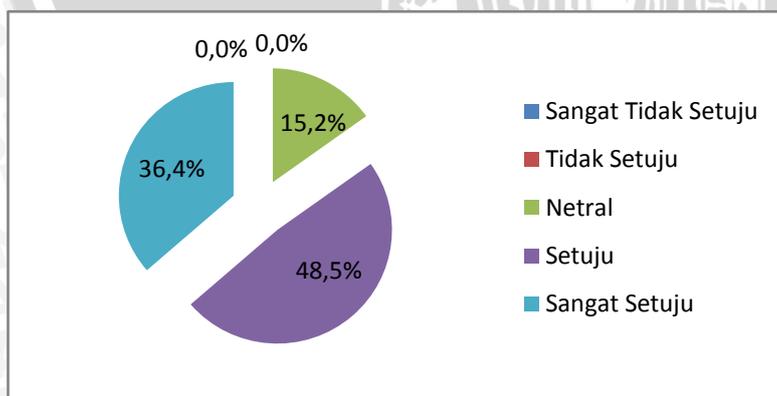
Pernyataan 1 : Saya merasa navigasi didalam aplikasi mudah bagi saya



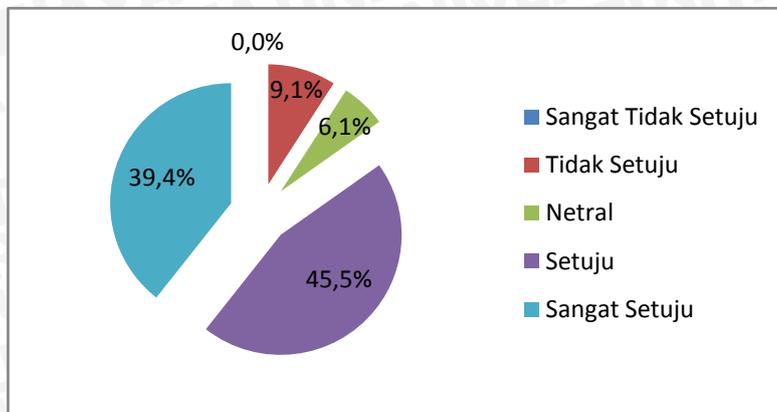
Pernyataan 2 : Saya merasa interaksi didalam aplikasi jelas dan mudah dimengerti



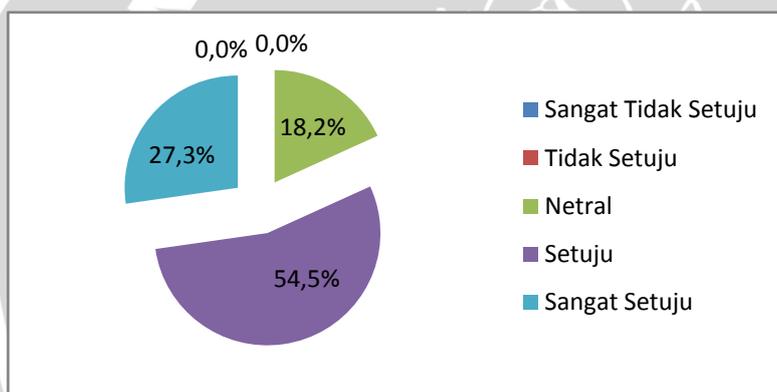
Pernyataan 3 : Saya merasa sangat mudah menggunakan aplikasi ini



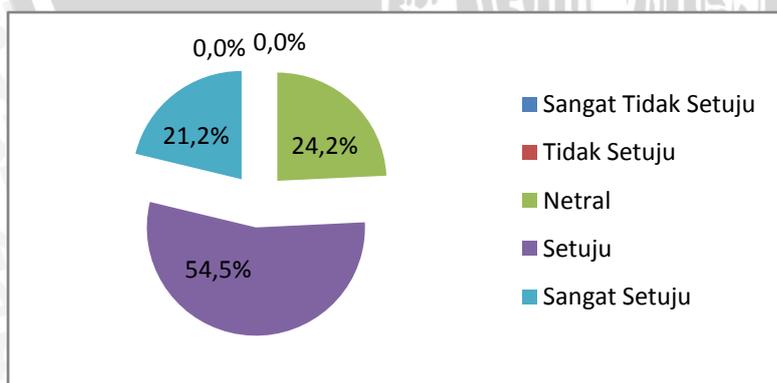
Pernyataan 4 : Saya merasa aplikasi ini memberikan respon kinerja yang cepat saat dioperasikan



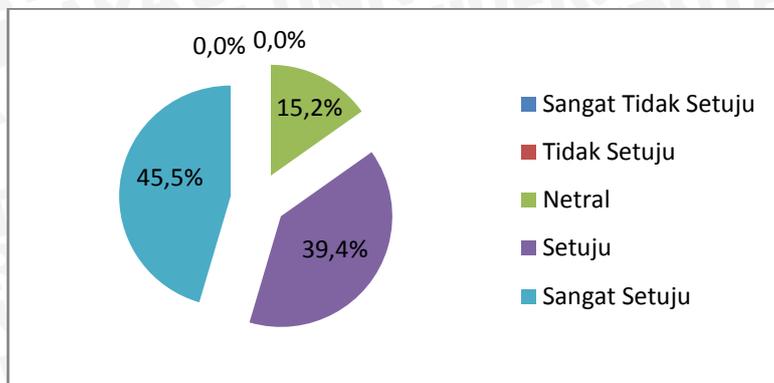
Pernyataan 5 : Saya merasa informasi yang ditampilkan pada aplikasi sangat jelas dan mudah dipahami



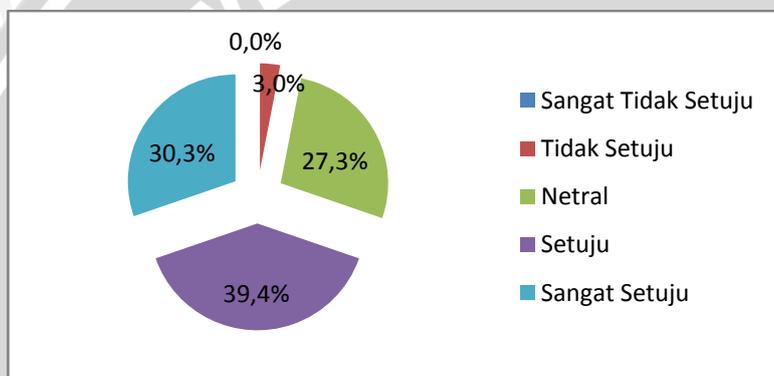
Pernyataan 6 : Aplikasi ini sangat berguna bagi saya



Pernyataan 7 : Saya merasa aplikasi ini dapat membantu saya untuk mengumpulkan informasi secara lengkap



Pernyataan 8 : Saya akan menggunakan aplikasi ini sesering mungkin



Pernyataan 9 : Saya merasa terbantu dengan adanya aplikasi ini dalam mengetahui semua informasi kebutuhan dalam melakukan budidaya secara mandiri

