

**RANCANG BANGUN WEBSITE SISTEM HITUNG CEPAT
(QUICKCOUNT) PADA PEMILIHAN UMUM**

SKRIPSI

**Diajukan Untuk Memenuhi Persyaratan Memperoleh Gelar Sarjana
Komputer**



Disusun Oleh:

DARA NISA' ULUM BILGIES

0910681005

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER**

MALANG

2014



LEMBAR PERSETUJUAN

**RANCANG BANGUN WEBSITE SISTEM HITUNG CEPAT
(QUICKCOUNT) PADA PEMILIHAN UMUM**

SKRIPSI

Diajukan Untuk Memenuhi Persyaratan Memperoleh Gelar Sarjana

Komputer



Disusun Oleh:

DARA NISA' ULUM BILGIES

NIM. 0910681005

Telah diperiksa dan disetujui :

Pembimbing I

Pembimbing II

Aryo Pinandito, S.T., M.MT.

NIK. 830519 16 1 1 0374

Barlian Henryranu Prasetio, S.T., M.T.

NIK. 821024 06 1 1 0254

LEMBAR PENGESAHAN

**RANCANG BANGUN WEBSITE SISTEM HITUNG CEPAT
(QUICKCOUNT) PADA PEMILIHAN UMUM**

Disusun Oleh :

DARA NISA' ULUM BILGIES

NIM. 0910681005

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 11 Juli 2014

Penguji I

Penguji II

Denny Sagita Rusdianto, S.Kom., M.Kom.

Aswin Suharsono, S.T., M.T.

NIK. 851124 06 1 1 0250

NIK. 840919 06 1 1 0251

Penguji III

Fajar Pradana, S.ST.,M.Eng.

NIK. 87112116110371

Mengetahui

Ketua Program Studi Teknik Informatika/Illmu Komputer

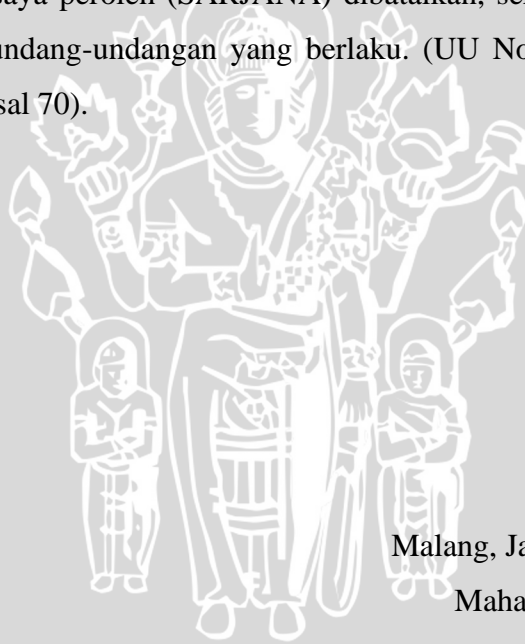
Drs. Marji, MT.

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, Januari 2014

Mahasiswa,

Dara Nisa' Ulum Bilgies

NIM. 091681005

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT karena berkat rahmat-Nya, penulis dapat menyelesaikan skripsi yang berjudul “**RANCANG BANGUN WEBSITE SISTEM HITUNG CEPAT (QUICKCOUNT) PADA PEMILIHAN UMUM**”.

Penyusunan skripsi ini bertujuan untuk memenuhi sebagian persyaratan memperoleh gelar sarjana komputer. Tak lupa penulis mengucapkan rasa terima kasih kepada :

1. Aryo Pinandito, S.T., M.MT. dan Barlian Henryranu Prasetyo, S.T., M.T. selaku dosen pembimbing selama pelaksanaan skripsi.
2. Ibunda Umu Rodhiyah, Ayahanda Abdul Fatah, S.Pd., M.Pd., Mas Budi Satria Maulana, S.E., Adik Farah Cahyawati Lazuardini, Adik Shefira Tsalits Nur A'izza dan seluruh keluarga atas segenap dukungan dan kasih sayang yang telah diberikan.
3. Ir. Sutrisno, M.T, Ir. Heru Nurwasito, M.Kom., Himawat Aryadita, S.T, M.Sc., dan Edy Santoso, S.Si., M.Kom., selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Drs. Marji, M.T dan Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
5. Seluruh Dosen dan Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya, bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
6. Ria Agustina, S.Kom., Kautsarani Permata alam, S.Kom., Melynda Aula, S.Kom., Alfina Zulfa Wahib, S.Kom., kalian nelongso mania penghibur disaat kegalauan akan skripsi.
7. Luthfi Aziz, S.Kom., Geri Arif Wicaksono, S.Kom., Alfi Aulia, S.Kom., Danial M. Zaki, S.Kom., Wahyu Mulya Candra, S.Kom., Meidi Dian, S.Kom.,

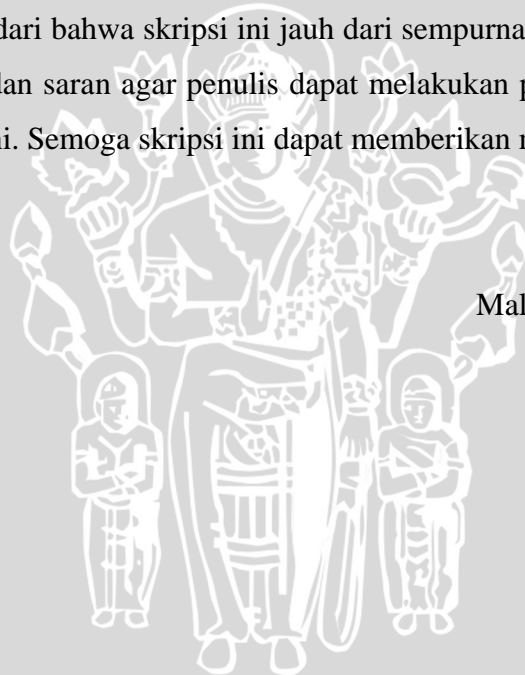
Hamdan Hamaris, S.Kom., Saiful Setyawan, S.Kom, Haryo Sarwo, S.Kom., dan semua teman D09 telah membantu selama perkuliahan.

8. Ruben, Mbak Awl, Gok Gigih, Lambeck, A' Ulum, Agus, Mas Anas, Eko Pecok, dan seluruh sahabat-sahabati Keluarga Besar PMII Komisariat Brawijaya yang telah mengizinkan saya untuk ikut belajar berorganisasi.
9. Teman-teman Averroes Community yang telah berbagi data untuk kelengkapan skripsi penulis.
10. Seluruh teman-teman TIF Angkatan 2009 yang menemani dan selalu memberikan bantuan baik moril maupun spiritual.
11. Seluruh pihak yang telah membantu kelancaran penulisan skripsi yang tidak dapat saya sebutkan satu persatu.

Penulis menyadari bahwa skripsi ini jauh dari sempurna. Untuk itu penulis mengharapkan kritik dan saran agar penulis dapat melakukan perbaikan terhadap skripsi yang disusun ini. Semoga skripsi ini dapat memberikan manfaat. Amiin.

Malang, Juni 2014

Penulis



ABSTRAK

Dara Nisa' Ulum Bilgies, 2014 : RANCANG BANGUN WEBSITE SISTEM HITUNG CEPAT (*QUICKCOUNT*) PADA PEMILIHAN UMUM. Peneliti Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dosen Pembimbing : Aryo Pinandito, S.T., M.MT. dan Barlian Henryranu Prasetio, S.T., M.T.

Perhitungan suara pada pemilihan umum dilakukan secara bertahap, yaitu dimulai dari proses perhitungan suara di tempat pemungutan suara (TPS) sampai proses perhitungan suara digabungkan menjadi satu dalam tempat yang sama. Proses perhitungan suara membutuhkan waktu yang lama, bisa mencapai seminggu bahkan sebulan. Hal ini dapat menyebabkan terjadi ketidakpastian informasi terhadap pemenang dalam pemilihan umum. Perhitungan suara cepat yang disebut *Quickcount* merupakan salah satu cara dalam melakukan perhitungan suara pada pemilihan umum dengan menggunakan metode perhitungan secara acak terhadap beberapa *sample* TPS yang telah ditentukan. Pengumpulan data hasil suara disetiap TPS *sample* dilakukan oleh beberapa relawan (*surveyor*) yang memantau langsung ketika perhitungan suara di TPS. Relawan (*surveyor*) akan melaporkan hasil suara tersebut ke pusat pengumpulan suara. Hal ini dibutuhkan sistem untuk menghubungkan beberapa *surveyor* agar data hasil suara dapat digabungkan menjadi satu. Sehingga *quickcount* hasil pemilihan umum dapat diketahui secara cepat oleh masyarakat melalui website sistem hitung cepat.

Kata Kunci : *Quickcount*, TPS, *Sampling*

ABSTRACT

Dara Nisa' Ulum Bilgies, 2014 : WEBSITE DESIGN SYSTEM QUICK COUNT AT GENERAL ELECTION. *Undergraduate Thesis of Informatic Study Program, Information Technology and Computer Science Program, Brawijaya University, Malang.*

Advisor: Aryo Pinandito, S.T., M.MT. and Barlian Henryranu Prasetyo, S.T., M.T.

Calculation of voice at general election conducted step by step, beginning of the process of vote counting at polling stations until the vote counting process are combined into one in the same place. vote counting process takes a long time, can reach one week or even one month. This can lead to the uncertainty information in the general election winner. Calculation of voice swiftly the so-called Quick Count represent one way of doing calculation of voice at general election by using calculation method at random to some polling stations sample which have been determined. Data collecting result of voice in every polling stations sample conducted by some volunteers which watching directly when counting of vote at polling stations. Volunteers will report the results of the vote to the polling center. This matter is required by system to connect some volunteers in order to data result of vote can be combine into one. So that quick count result of general election can be seen quickly by public through the website quick count.

Keyword : Quickcount, TPS, Sampling



DAFTAR ISI

KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1. LATAR BELAKANG MASALAH	1
1.2. RUMUSAN MASALAH	2
1.3. BATASAN MASALAH	2
1.4. TUJUAN PENELITIAN	3
1.5. MANFAAT PENELITIAN	3
1.6. SISTEMATIKA PENULISAN	3
BAB II DASAR TEORI	5
2.1 Quickcount	5
2.2 Unified Modelling Language	7
2.2.1. Use case Diagram	7
2.2.2. Class Diagram	8
2.2.3. Activity Diagram	9
2.2.4. Sequence diagram	10
2.3 PHP – <i>CodeIgniter Framework</i>	11
2.4 Model View Controller (MVC)	12
2.5 Apache Web Server	13
2.6 Pengujian Perangkat Lunak	13
BAB III METODOLOGI PENELITIAN	15
3.1 Studi Literatur	15
3.2 Analisa Kebutuhan	16
3.3 Perancangan Perangkat Lunak	16
3.4 Implementasi Perangkat Lunak	17
3.5 Pengujian Perangkat Lunak	17



3.6	Pengambilan Kesimpulan dan Saran.....	18
BAB IV PERANCANGAN PERANGKAT LUNAK.....		19
4.1	Analisis Kebutuhan	19
4.1.1	Identifikasi Aktor.....	19
4.1.2	Identifikasi Kebutuhan	20
4.2	Perancangan Sistem Perangkat Lunak	20
4.2.1	Perancangan <i>Use Case</i>	21
4.2.2	Perancangan Activity Diagram.....	23
4.2.3	Perancangan <i>Sequence Diagram</i>	27
4.2.4	Perancangan Class Diagram	31
4.2.5	Perancangan Antarmuka.....	32
BAB V IMPLEMENTASI PERANGKAT LUNAK.....		35
5.1	Spesifikasi Lingkungan Sistem	35
5.1.1	Spesifikasi Lingkungan Perangkat Keras	35
5.1.2	Spesifikasi Lingkungan Perangkat Lunak	35
5.2	Batasan Implementasi Sistem.....	36
5.3	Implementasi Basis Data.....	36
5.4	Implementasi <i>File Program</i>	37
5.5	Implementasi Antarmuka	39
5.5.1	Antarmuka bagi <i>Viewer</i>	39
5.5.2	Antarmuka bagi <i>Surveyor</i>	40
BAB VI PENGUJIAN DAN ANALISIS		41
6.1	Pengujian.....	41
6.1.1	Pengujian Validasi	41
6.1.2	Pengujian Non Fungsional.....	47
6.2	Analisis.....	47
6.2.1	Analisis Pengujian Validasi	47
6.2.2	Analisis Pengujian Non Fungsional.....	48
BAB VII PENUTUP.....		49
7.1.	KESIMPULAN	49
7.2.	SARAN	49



DAFTAR PUSTAKA DP-1

LAMPIRAN L-1



DAFTAR GAMBAR

Gambar 2. 1 Contoh Use Case Diagram.....	8
Gambar 2. 2 Notasi UML untuk Relasi Asosiasi.....	9
Gambar 2. 3 Notasi UML untuk Relasi Aggregation	9
Gambar 2. 4 Contoh Activity Diagram.....	10
Gambar 2. 5 Contoh Sequence Diagram.....	10
Gambar 2. 6 Struktur File CI	12
Gambar 3. 1 Metodologi Penelitian	15
Gambar 4. 1 Use Case Diagram.....	21
Gambar 4. 2 Activity diagram olah data hasil quickcount.....	25
Gambar 4. 3 Activity Diagram lihat hasil quickcount	26
Gambar 4. 4 Sequence diagram olah data hasil quickcount	28
Gambar 4. 5 Sequence diagram Lihat Hasil Quickcount.....	30
Gambar 4. 6 Perancangan Class Diagram Olah Data Hasil Quickcount	32
Gambar 4. 7 Perancangan Antarmuka halaman utama/viewer	33
Gambar 4. 8 Perancangan Antarmuka halaman input hasil oleh surveyor	34
Gambar 5. 1 Basis Data Sistem Quickcount.....	36
Gambar 5. 2 Implementasi file program Controler tb_hasil.php	37
Gambar 5. 3 Implementasi file program Models_hasil.php.....	38
Gambar 5. 4 Implementasi file program List_tb_hasil.tpl.....	38
Gambar 5. 5 Implementasi Antarmuka bagi Viewer	39
Gambar 5. 6 Implementasi Antarmuka Input Data oleh Surveyor	40

DAFTAR TABEL

Tabel 4. 1 Identifikasi Aktor	19
Tabel 4. 2 Daftar Kebutuhan Fungsional	20
Tabel 4. 3 Use Case Olah Data Hasil Quickcount	21
Tabel 4. 4 Use Case Lihat Hasil Quickcount	23
Tabel 5. 1 Spesifikasi Lingkungan Perangkat Keras	35
Tabel 5. 2 Spesifikasi Lingkungan Perangkat Lunak	35
Tabel 6. 1 Kasus Uji Validasi Penambahan Data Hasil Quickcount	41
Tabel 6. 2 Kasus Uji Validasi Penambahan Data Hasil Quickcount Non Ideal	43
Tabel 6. 2 Kasus Uji Validasi Penghapusan Data Hasil Quickcount.....	44
Tabel 6. 3 Kasus Uji Validasi Pengubahan Data Hasil Quickcount	45
Tabel 6. 4 Kasus Uji Validasi Menampilkan Halaman Hasil Perolehan Suara <i>Quickcount</i>	46



DAFTAR LAMPIRAN

Lampiran 1. Deskripsi Perancangan Use Case loginL-1

Lampiran 2 Deskripsi Perancangan Use Case Olah UserL-2

Lampiran 3 Deskripsi Perancangan Use Case Olah KandidatL-3

Lampiran 4 Deskripsi Perancangan Use Case KecamatanL-4

Lampiran 5 Deskripsi Perancangan Use Case PeriodeL-5

Lampiran 6 Deskripsi Perancangan Use Case TPSL-6

Lampiran 7 Deskripsi Perancangan Use Case Lihat Log Quickcount.....L-7

Lampiran 8 Activity Diagram LoginL-8

Lampiran 9 Activity Diagram Olah User.....L-9

Lampiran 10 Activity Diagram Olah Kandidat.....L-10

Lampiran 11 Activity Diagram KecamatanL-11

Lampiran 12 Activity Diagram Periode.....L-12

Lampiran 13 Activity Diagram TPSL-13

Lampiran 14 Activity Diagram Log Quickcount.....L-14

Lampiran 15 Sequence diagram LoginL-15

Lampiran 16 Sequence diagram Olah UserL-16

Lampiran 17 Sequence diagram Olah KandidatL-17

Lampiran 18 Sequence diagram Kecamatan.....L-18

Lampiran 19 Sequence diagram PeriodeL-19

Lampiran 20 Sequence diagram TPSL-20

Lampiran 21 Sequence diagram Log Quickcount.....L-21

Lampiran 22 Perancangan Antarmuka LoginL-22

Lampiran 23 Perancangan Antarmuka Home Admin.....L-23

Lampiran 24 Perancangan Antarmuka Olah Data AdminL-24

Lampiran 25 Perancangan Antarmuka Home SurveyorL-25

Lampiran 26 Data Quickcount Pemilihan Walikota Malang Tahun 2013.....L-26

Lampiran 27 Kasus Uji Validasi Penambahan Data Kecamatan.....L-31

Lampiran 28 Kasus Uji Validasi Penghapus Data KecamatanL-32

Lampiran 29 Kasus Uji Validasi Pengubahan Data Kecamatan.....L-33



Lampiran 30 Kasus Uji Validasi Penambahan Data Periode.....	L-34
Lampiran 31 Kasus Uji Validasi Penghapus Data Periode.....	L-35
Lampiran 32 Kasus Uji Validasi Pengubahan Data Periode.....	L-36
Lampiran 33 Kasus Uji Validasi Penambahan Data TPS	L-37
Lampiran 34 Kasus Uji Validasi Penghapus Data TPS	L-38
Lampiran 35 Kasus Uji Validasi Pengubahan Data TPS.....	L-39
Lampiran 36 Kasus Uji Validasi Penambahan Data Kandidat	L-40
Lampiran 37 Kasus Uji Validasi Penghapusan Data Kandidat.....	L-41
Lampiran 38 Kasus Uji Validasi Pengubahan Data Kandidat	L-42
Lampiran 39 Kasus Uji Validasi Penambahan Data User	L-43
Lampiran 40 Kasus Uji Validasi Penghapusan Data User.....	L-44
Lampiran 41 Kasus Uji Validasi Pengubahan Data User	L-45
Lampiran 42 Kasus Uji Validasi Menampilkan Data Log Quickcount.....	L-46



BAB I PENDAHULUAN

1.1. LATAR BELAKANG MASALAH

Dalam sistem pemilihan umum (pemilu) terjadi banyak masalah yang dihadapi oleh berbagai pihak. Permasalahan yang sering terjadi adalah dalam pelaksanaan perhitungan suara yang secara manual sehingga membutuhkan waktu satu minggu. Sehingga banyak berbagai pihak mulai dari kalangan lembaga swadaya masyarakat (LSM), Pengamat Politik, dan masyarakat menanti akan hasil suara terbanyak. Sebagian kalangan menggunakan alat teknologi untuk sekedar saling memberi info tentang kabar suara terbanyak di tempat pemungutan suara (TPS) masing-masing. Dalam hal ini mengakibatkan terjadi kesalahan, sehingga beberapa pihak lembaga swadaya masyarakat (LSM) ikut berpartisipasi melaksanakan perhitungan suara. Perhitungan suara dibagi menjadi dua, perhitungan cepat (*quick count*) dan perhitungan nyata (*real count*). Perhitungan cepat (*quick count*) merupakan beberapa sample data yang telah dihitung dengan pemetaan politik yang sesuai dan kemudian dilakukan perhitungan suara. Perhitungan nyata (*Real count*) merupakan hasil perhitungan suara di semua tempat pemungutan suara.

Quick count telah diterapkan di Indonesia sejak 1997 oleh LP3ES (Lembaga Pelatihan, Penelitian, Penerangan, Ekonomi dan Sosial) pada pemilu terakhir rezim Soeharto yang dilakukan secara diam-diam bekerjasama dengan salah satu kekuatan politik. *Quick count* ini cukup berhasil, dengan satu hari setelah pelaksanaan pemilu LP3ES mampu memprediksi hasil pemilu di DKI Jakarta persis sebagaimana hasil perhitungan suara oleh LPU (Lembaga Pemilihan Umum). LP3ES secara akurat berhasil memprediksi pemenang pemilu dan komposisi pemenang pemilu dari urutan 1 sampai 24. [KSM-07]

Metode *quick count* dengan menggunakan teknik *multistage random sampling* merupakan metode dimana setiap anggota populasi akan mendapatkan peluang yang sama untuk terpilih sebagai sampel. Model pengacakan dilakukan secara bertingkat yang dilakukan secara proporsional berdasarkan besaran populasi jumlah TPS. Dengan metode pengacakan seperti di atas, maka akan

didapatkan jumlah sampel yang proporsional sesuai dengan prosentase jumlah TPS di daerah tersebut sehingga hasil quick count dapat digeneralisasi dan mencerminkan hasil pemilu dengan akurasi tinggi. [LSI-JIP]

Ditahun 2013 di Lembaga Swadaya Masyarakat (LSM) Averroes Community masih melakukan perhitungan cepat (*quick count*) dengan metode yang sederhana. Dengan menggunakan banyak pemantau (*surveyor*) relawan untuk mencatat informasi di lokasi pemungutan suara dan membutuhkan banyak admin guna mengelolah data yang dikirim oleh pemantau (*surveyor*) melalui SMS (*Short Message Service*) kemudian di-*input* ulang secara manual dengan pengolahan data maupun website yang digunakan. Dan pada saat periode pemilihan suara selesai, pihak LSM akan membuat kembali sistem tersebut dikemudian hari. Dalam hal ini akan memerlukan waktu minimal seminggu untuk mempersiapkan dan melakukan *quick count*.

Mengingat pentingnya keefektifan waktu, maka diperlukan suatu sistem agar kegiatan *quick count* dapat terselenggara secara cepat dan terdokumentasikan sebaik mungkin. Sistem dirancang bersifat dinamis sehingga data/informasi dapat diakses secara *real time*. Dengan bahasa pemrograman PHP dengan *framework CodeIgniter*, database MySQL, dan pendekatan pengembangan sistem secara MVC (*Model, View, Controller*) memungkinkan untuk dilakukannya pengembangan sistem secara dinamis dan berorientasi objek.

1.2. RUMUSAN MASALAH

Berdasarkan pada permasalahan yang telah dijelaskan pada bagian latar belakang, maka rumusan masalah dapat disusun: “Bagaimana cara membuat proses quickcount menjadi lebih efisien dengan menggunakan sistem berbasis web?”.

1.3. BATASAN MASALAH

Ruang lingkup dalam penelitian ini berdasarkan sebagai berikut:

- a. Perhitungan cepat (*quick count*) menggunakan data quickcount pemilihan walikota Malang pada tahun 2013 dari salah satu LSM di Kota Malang.
- b. Data *quickcount* dianggap sebagai data asli hasil perhitungan suara.

- c. Pengujian sistem dilakukan dengan menggunakan pengujian sistem secara fungsional dalam bentuk fungsional sistem dan non fungsional dalam bentuk kuisioner oleh pengguna sistem.

1.4. TUJUAN PENELITIAN

Tujuan dari penelitian ini adalah membuat sistem untuk pengolahan data perhitungan suara dengan metode *quick count*.

1.5. MANFAAT PENELITIAN

Manfaat yang bisa didapatkan dari penelitian ini adalah :

1. Mendapatkan sistem yang berguna untuk LSM dalam bidang hitung cepat pada pemilihan umum.
2. Menyediakan alternatif pengembangan sistem untuk perhitungan suara dengan metode hitung cepat
3. Memudahkan LSM untuk melakukan pengarsipan dan perhitungan suara disetiap periodenya.
4. Untuk mempermudah pengaksesan sehingga mendapatkan informasi terbaru.

1.6. SISTEMATIKA PENULISAN

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II Dasar teori

Menguraikan teori dasar dan teori penunjang yang berkaitan dengan perancangan sistem hitung cepat (*quick count*) pada pemilihan umum yang berbasis bahasa pemrograman PHP dan *database* MySQL.

BAB III Metodologi Penelitian

Bagian ini membahas langkah-langkah yang akan ditempuh dalam penulisan skripsi dengan menggunakan bahasa pemrograman PHP dan *database* MySQL.

BAB IV Analisis dan Perancangan Sistem

Membahas analisis kebutuhan dan perancangan sistem serta hasil yang diharapkan.

BAB V Implementasi

Membahas tentang implementasi dari sistem.

BAB VI Pengujian Sistem

Memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

BAB VII Penutup

Memuat kesimpulan serta saran yang diperoleh dari pembuatan dan pengujian sistem untuk pengembangan lebih lanjut.



BAB II

DASAR TEORI

Bab ini membahas dasar-dasar teori yang digunakan untuk menunjang penulisan skripsi “Rancang Bangun Website Sistem Hitung Cepat (Quickcount) pada Pemilihan Umum”. Dasar teori yang diperlukan berdasarkan latar belakang dan rumusan masalah adalah : *Quickcount*, UML, PHP, MVC, *CodeIgniter*, HTML, *Apache Server*, dan pengujian perangkat lunak.

2.1 Quickcount

Real Count merupakan sebuah proses pengumpulan informasi oleh banyak relawan melalui pemantauan langsung saat pemungutan dan perhitungan suara di seluruh tempat pemungutan suara (TPS) yang ada. Pemantau mencatat informasi, termasuk hasil perhitungan suara yang ada, dan melaporkan hasil tersebut ke pusat pengumpulan data melalui telepon. Dalam permasalahan perhitungan suara *real count*. Pemilu yang membutuhkan waktu beberapa hari atau bahkan beberapa minggu. [ERI-06]

Quickcount atau penghitungan cepat adalah proses pencatatan hasil perolehan suara di ribuan Tempat Pemungutan Suara (TPS) yang dipilih secara acak (Ujiyati 2004).

Quickcount merupakan prediksi hasil pemilu berdasarkan fakta bukan berdasarkan opini. Sehingga *quickcount* tidak sama dengan jajak pendapat terhadap pemilih. (Kristanto 2007).

Penentuan besaran sampel pada quick count didasarkan oleh derajat keragaman(variability), margin of error (MoE) dan tingkat kepercayaan (confidence interval). Istilah MoE sering disamaartikan dengan pengertian sampling error , dimana sebenarnya sampling error dihitung setelah survei selesai dilakukan sesuai dengan teknik sampling yang digunakan. [ESTOK-2002].

Tahapan proses *quickcount* secara singkat menurut LSI & JIP (2007) adalah

1. Menentukan besar sampel pemilih yang akan diamati

$$n = \frac{Z^2 \cdot [p(1-p)] \cdot N}{Z^2 \cdot [p(1-p)] + (N-1) \cdot E^2}$$

Keterangan :

n = Besar sampel pemilih

Z = Nilai distribusi normal pada tingkat kepercayaan yang dipakai.

Tingkat kepercayaan yang dipakai 90% nilai Z adalah 1,65.

Tingkat keercayaan yang dipakai 95% nilai Z adalah 1,96. Tingkat kepercayaan 99% nilaai Z adalah 2,58.

p = Variasi populasi dalam bentuk proporsi. Proporsi dibagi dalam dua bagian dengan total 100%. Proporsi yang digunakan adalah pada saat keragaman tertinggi terjadi dimana $p = 50\%$ (atau 0.5).

E = Kesalahan sampling yang dikehendaki (sama dengan *margin of error*)

N = Jumlah populasi pemilih

- Memilih TPS yang akan diamati secara acak

$$t = \frac{n}{\text{rata - rata pemilih per TPS}}$$

Keterangan :

t = Jumlah TPS yang akan diambil sample

n = Besar Sample pemilih

$$\text{rata - rata pemilih per TPS} = \frac{\text{Jumlah pemilih}}{\text{banyak TPS}}$$

Memilih TPS secar acak dilakukan dengan menghitung interval sample

dengan cara : $\frac{\text{Jumlah TPS}}{\text{jumih TPS sample}}$

- Manajemen data (pengamatan, pencatatan, dan analisa data hasil perhitungan suara)

- Publikasi hasil *quickcount*

Prediksi *quickcount* akan akurat apabila mengacu pada metodologi statistik dan penarikan sampel yang ketat serta diimplementasikan secara konsisten di lapangan. Kekuatan *quickcount* juga sangat tergantung pada

identifikasi terhadap berbagai faktor yang berdampak pada distribusi suara dalam populasi pemilih. Apabila pemilu berjalan lancar tanpa kecurangan, akurasi *quickcount* dapat disandarkan pada perbandingannya dengan hasil resmi KPU. Tetapi apabila pemilu berjalan penuh kecurangan, maka hasil *quickcount* dapat dikatakan kredibel meskipun hasilnya berbeda dengan hasil resmi KPU. [Kis-2007]

Quick count lahir dari kebutuhan untuk menjaga agar penghitungan suara pemilu atau Pilkada tidak curang. Kalaupun ada kecurangan, diharapkan tidak merubah siapa yang seharusnya menang atau kalah. Quick count menjaga suara pemilih dan membantu agar proses pemilu atau Pilkada berlangsung secara jujur. [SMJ-08]

2.2 Unified Modelling Language

UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat software berorientasi objek. Karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma *object oriented*. [PM-03]

UML mendefinisikan diagram-diagram yang memiliki fungsi dan informasi yang berbeda-beda. Berikut beberapa diagram yang dipergunakan dalam perancangan sistem ini:

2.2.1. Use case Diagram

Use case diagram menggambarkan unit fungsionalitas yang disediakan oleh sistem. Tujuan utama dari diagram *use case* adalah untuk membantu pengembangan memvisualisasikan persyaratan fungsional dari sistem, termasuk hubungan "aktor" (manusia yang akan berinteraksi dengan sistem) proses penting, serta hubungan antara penggunaan yang berbeda kasus.

Diagram *Use Case* berguna dalam tiga hal :

1. Menjelaskan fasilitas yang ada (*requirements*)

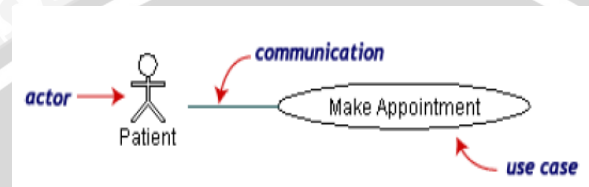
Use Case baru selalu menghasilkan fasilitas baru ketika sistem dianalisa, dan design menjadi lebih jelas.

2. Komunikasi dengan klien

Penggunaan notasi dan simbol dalam diagram *Use Case* membuat pengembang lebih mudah berkomunikasi dengan kliennya.

3. Membuat test dari kasus-kasus secara umum

Kumpulan dari kejadian-kejadian untuk *Use Case* bisa dilakukan test kasus layak untuk kejadian-kejadian tersebut.



Gambar 2. 1 Contoh Use Case Diagram

2.2.2. Class Diagram

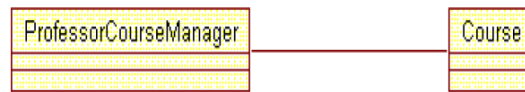
Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan pada atribut atau properti suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, package dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

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

Diagram *class* mempunyai 3 macam *relationships* (hubungan) yaitu:

a. Association

Suatu hubungan antara bagian dari dua kelas. Terjadi association antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Di dalam diagram sebuah association adalah penghubung yang menghubungkan dua kelas seperti pada Gambar 2.2.



Gambar 2. 2 Notasi UML untuk Relasi Asosiasi.

b. *Aggregation*

Aggregation merupakan suatu *association* dimana salah satu kelasnya merupakan bagian dari suatu kumpulan. *Aggregation* memiliki titik pusat yang mencakup keseluruhan bagian. Pada Gambar 2.3 merupakan notasi UML untuk relasi aggregation.



Gambar 2. 3 Notasi UML untuk Relasi Aggregation

c. *Generalization*

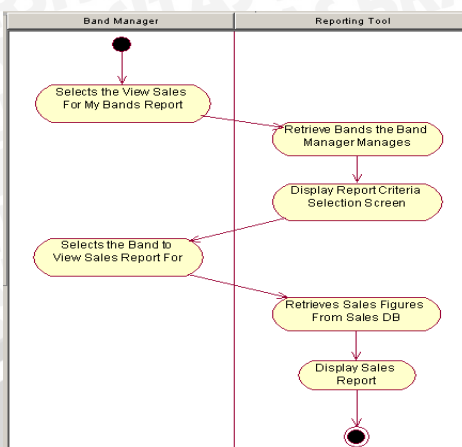
Suatu hubungan turunan dengan mengasumsikan satu kelas merupakan suatu *superClass* (kelas super) dari kelas yang lain. *Generalization* memiliki tingkatan yang berpusat pada *superClass*.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja

2.2.3. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). *Activity diagram*nya menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

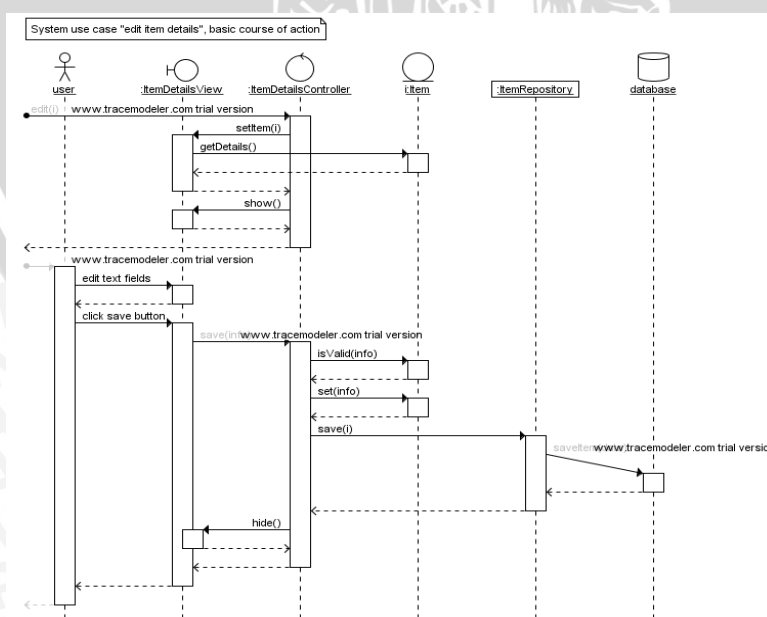


Gambar 2. 4 Contoh Activity Diagram

2.2.4. Sequence diagram

Sequence diagram menunjukkan aliran rinci untuk kasus penggunaan tertentu atau bahkan hanya bagian dari kasus penggunaan tertentu. Pada Gambar 2.5 merupakan contoh *sequence diagram*. Sebuah *sequence diagram* memiliki dua dimensi:

1. Dimensi vertikal menunjukkan urutan pesan / panggilan dalam urutan waktu yang mereka terjadi.
2. Dimensi horizontal menunjukkan contoh objek yang pesan dikirim.



Gambar 2. 5 Contoh Sequence Diagram

2.3 PHP – *CodeIgniter Framework*

Script PHP (*Hypertext Preprocessor*) adalah bahasa program yang berjalan pada sebuah webserver atau sering disebut *server-side*. *Server-side* adalah agar dapat menjalankan script ini dibutuhkan tiga program utama, yaitu web-server (dapat berupa IIS dari windows atau apache), modul PHP dan juga web browser. PHP dapat berjalan pada beberapa jenis web-server antara lain *Apache, Microsoft IIS, personal webserver, Netscape dan Iplanet Server, Caudium, Xitami, Omnihttpd* dan masih banyak lagi. [ST-05]

Ada beberapa macam *framework* yang digunakan dalam bahasa pemrograman PHP, diantaranya:

- a. *Cake PHP*
- b. *CodeIgniter*
- c. *Symfony*
- d. *Zend*
- e. *Yii*
- f. *Kohana*

CodeIgniter (CI) adalah sebuah *web application framework* yang bersifat *open source* untuk membangun aplikasi php dinamis. Tujuan utama pengembangan *CodeIgniter* adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat dan praktis daripada menulis semua *code* dari awal. *CodeIgniter* menyediakan berbagai macam *library* yang dapat mempermudah dalam pengerjaan dan pengembangan aplikasi. [SB-12]

CodeIgniter sangat ringan, terstruktur, dan mudah dipelajari. Selain itu *CodeIgniter* juga memiliki fitur – fitur lainnya yang sangat bermanfaat, antara lain:

- a. Menggunakan pattern MVC

Dengan menggunakan pattern MVC ini, struktur kode yang dihasilkan menjadi lebih terstruktur dan memiliki standar yang jelas.

- b. URL Friendly

URL yang dihasilkan sangat *url friendly*. Pada *CodeIgniter* diminimalisasi penggunaan `$_GET` dan digantikan dengan URL.

- c. Kemudahan

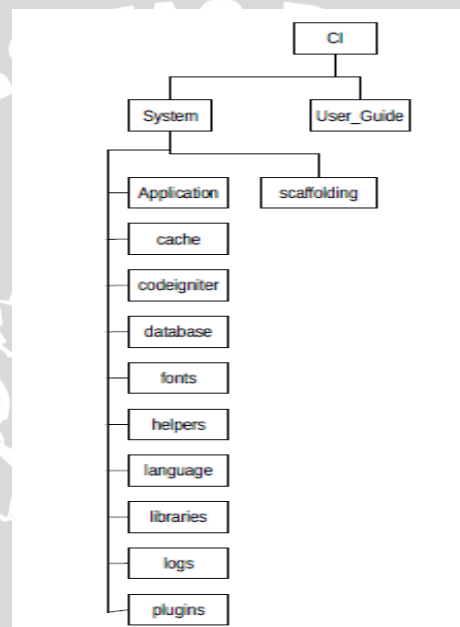
Kemudahan dalam mempelajari, membuat *library* dan *helper*, memodifikasi serta meng-integrasikan *library* dan *helper*.

d. Mudah dimodifikasi

Sangat mudah memodifikasi *behavior framework* ini. Tidak membutuhkan *server requirement* yang macam-macam serta mudah mengadopsi *library* lainnya.

e. Dokumentasi lengkap dan jelas

CodeIgniter telah menyediakan sebuah panduan yang lengkap didalamnya.



Gambar 2. 6 Struktur File CI

2.4 Model View Controller (MVC)

CodeIgniter menggunakan konsep MVC serta menyediakan banyak *library* dan *helper* untuk digunakan. MVC adalah sebuah *pattern* atau teknik pemrograman yang mempunyai tujuan utama untuk memisahkan bisnis *logic* (alur pikir), data *logic* (penyimpanan data) dan *presentation logic* (antarmuka aplikasi) atau secara sederhana yaitu memisahkan antara desain, data dan proses. Sehingga diharapkan dapat memudahkan proses modifikasi suatu bagian tanpa mempengaruhi bagian yang lainnya. Dalam MVC, model akan menjadi representasi data atau informasi dari sebuah aplikasi. View akan menjadi komponen yang berkaitan dengan tampilan. Dan controller akan menjadi pengatur

komunikasi yang terjadi dan juga aturan-aturan proses untuk memanipulasi data dari dan menuju model, yang kemudian ditampilkan dalam view.

2.5 Apache Web Server

Apache Web server adalah software yang memberikan layanan website. Web server menggunakan protokol yang disebut dengan HTTP (*HyperText Transfer Protocol*). Banyak pilihan *web server* di dunia *open source* yang dapat digunakan, penggunaannya bergantung pada keperluan. Salah satu web server yang sangat terkenal dan menjadi standar *de facto* setiap distribusi Linux, yaitu Apache. Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basisdata dan lain-lain.

2.6 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses atau serangkaian proses yang dirancang untuk memastikan kode komputer melakukan apa yang dirancang, sehingga kode dapat melakukan apa yang seharusnya dan tidak seharusnya dilakukan. Perangkat lunak harus diprediksi dan konsisten, sehingga tidak menimbulkan aksi yang diluar dugaan [GJM-04].

Pengujian perangkat lunak dapat menggunakan pendekatan test. Pendekatan test memunyai dua macam metode, yaitu :

1. *Black Box Testing*

Test case ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan dan apakah informasi yang disimpan secara eksternal selalu dijaga kemutakhirannya.

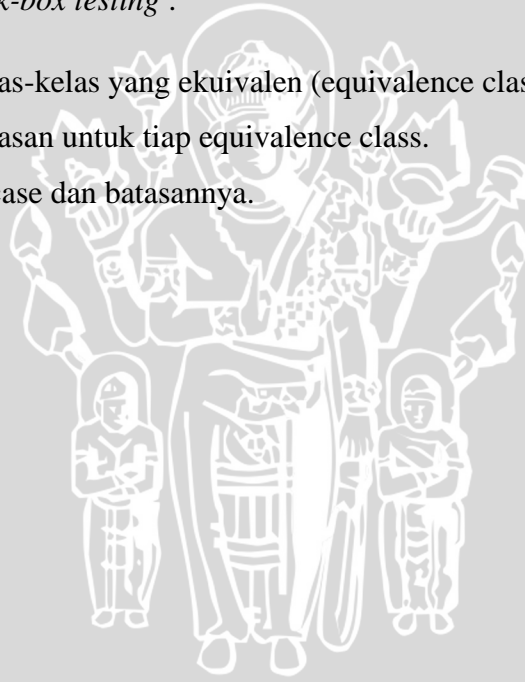
2. *White Box Testing*

Test case ini meramalkan cara kerja perangkat lunak secara rinci, karenanya logikal path (jalur logika) perangkat lunak akan ditest dengan menyediakan test case yang akan mengerjakan kumpulan kondisi dan atau pengulangan secara spesifik. Secara sekilas dapat diambil kesimpulan white box testing merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

Blackbox-testing menurut IEEE (1990) merupakan pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada output yang dihasilkan yang merespon input yang dipilih dan kondisi eksekusi. Pengujian *blackbox* dilakukan untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional tertentu dan untuk menguji fungsi-fungsi khusus dari PL yang dirancang. Kebenaran pengujian dilihat dari keluaran yang dihasilkan dari data atau kondisi masukan yang diberikan untuk fungsi yang ada tanpa melihat bagaimana proses untuk mendapatkan keluaran tersebut. Dari keluaran yang dihasilkan, kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahannya.

Langkah-langkah *black-box testing* :

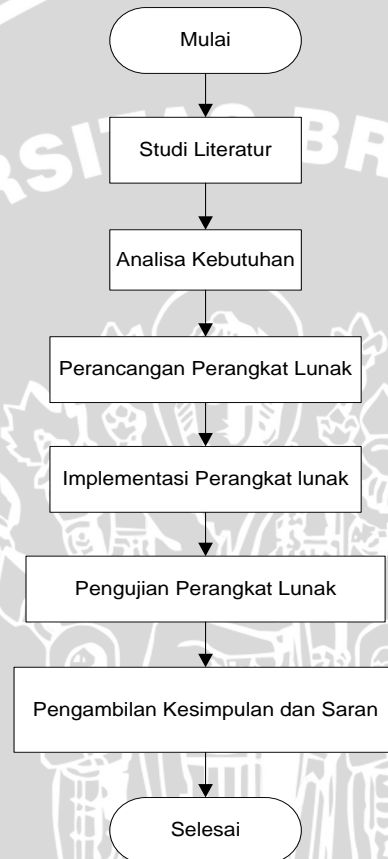
1. Identifikasi kelas-kelas yang ekuivalen (equivalence class).
2. Identifikasi batasan untuk tiap equivalence class.
3. Membuat test case dan batasannya.



BAB III

METODOLOGI PENELITIAN

Pada bab ini dijelaskan mengenai langkah-langkah yang akan ditempuh dalam penulisan skripsi, yaitu studi literatur, analisa kebutuhan, perancangan perangkat lunak, implementasi, pengujian sistem, dan penulisan laporan.



Gambar 3. 1 Metodologi Penelitian

Pada Gambar 3.1 merupakan gambar alur untuk melakukan proses penelitian.

3.1 Studi Literatur

Dalam perancangan dan implementasi penelitian ini perlu penelusuran literatur yang bertujuan sebagai dasar teori penunjang penulisan skripsi. Pengkajian teori-teori pendukung tersebut diperoleh dari buku, jurnal, e-book, laporan penelitian, website, maupun artikel. Literatur yang diperlukan untuk menunjang pengembangan sistem meliputi hal berikut :

- a. Sistem terdahulu, dimana masih menggunakan banyak admin dan banyak surveyor.
- b. *Quick count*
- c. PHP dengan *framework CodeIgniter*
- d. MySQL
- e. MVC
- f. Rekayasa perangkat lunak

3.2 Analisa Kebutuhan

Analisa kebutuhan bertujuan untuk menganalisa dan memperoleh semua informasi maupun data-data kebutuhan yang diperlukan dari sebuah sistem perangkat lunak yang akan dibangun. Hal pertama yang akan dilakukan dalam menganalisa kebutuhan adalah mengidentifikasi aktor sebagai pemeran di sistem tersebut. Dalam hal ini aktor yang berinteraksi di sistem ini adalah *administrator*, *surveyor*, *viewer*. Identifikasi kebutuhan merupakan langkah selanjutnya setelah penentuan aktor, meliputi identifikasi layanan yang disediakan oleh sistem dan identifikasi layanan pendukung yang diperlukan oleh sistem. Dalam melakukan identifikasi, hasil didapatkan berupa informasi maupun data yang kemudian digunakan untuk pembuatan *use case diagram*. Pembuatan *use case diagram* meliputi kegiatan yang dapat dilakukan dalam sistem, yaitu :

- a. *Administrator* dapat melakukan pengolahan data *surveyor*, pengolahan data kandidat calon dan *quick count*.
- b. *Surveyor* dapat melakukan *login* dan melakukan proses *input* hasil *quick count* yang telah ditentukan oleh administrator sebelumnya.
- c. *Viewer* hanya dapat melihat hasil perolehan suara.

3.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan setelah mendapatkan semua kebutuhan dari proses analisis kebutuhan. Perancangan sistem dibangun dengan menggunakan pendekatan berorientasi objek yang terdiri dari :

1. Perancangan *use case diagram* untuk aktor-aktor dan kegiatan apa saja yang akan dilakukan dalam sistem.

2. Perancangan kelas-kelas dan *interface* sehingga terhubung secara logika dengan *class diagram*.
3. Perancangan *activity diagram* untuk menggambarkan alur kegiatan sistem
4. Perancangan *sequence diagram* untuk menggambarkan scenario dari sebuah kejadian antar objek pada sistem sehingga mendapatkan keluaran tertentu
5. Perancangan tampilan antar muka pengguna sistem, yaitu :
 - a. Antar muka pengguna sistem *administrator*.
 - b. Antar muka pengguna sistem *surveyor*.
 - c. Antar muka pengguna sistem *viewer*.

3.4 Implementasi Perangkat Lunak

Implementasi perangkat lunak akan dilakukan setelah perancangan perangkat lunak selesai dilakukan. Implementasi perangkat lunak dilakukan dengan mengacu kepada fungsionalitas yang dirancang untuk membantu pengolahan data perhitungan suara dengan proses hitung cepat (*quick count*). Implementasi perangkat lunak ini menggunakan bahasa pemrograman PHP dan *database MySQL*.

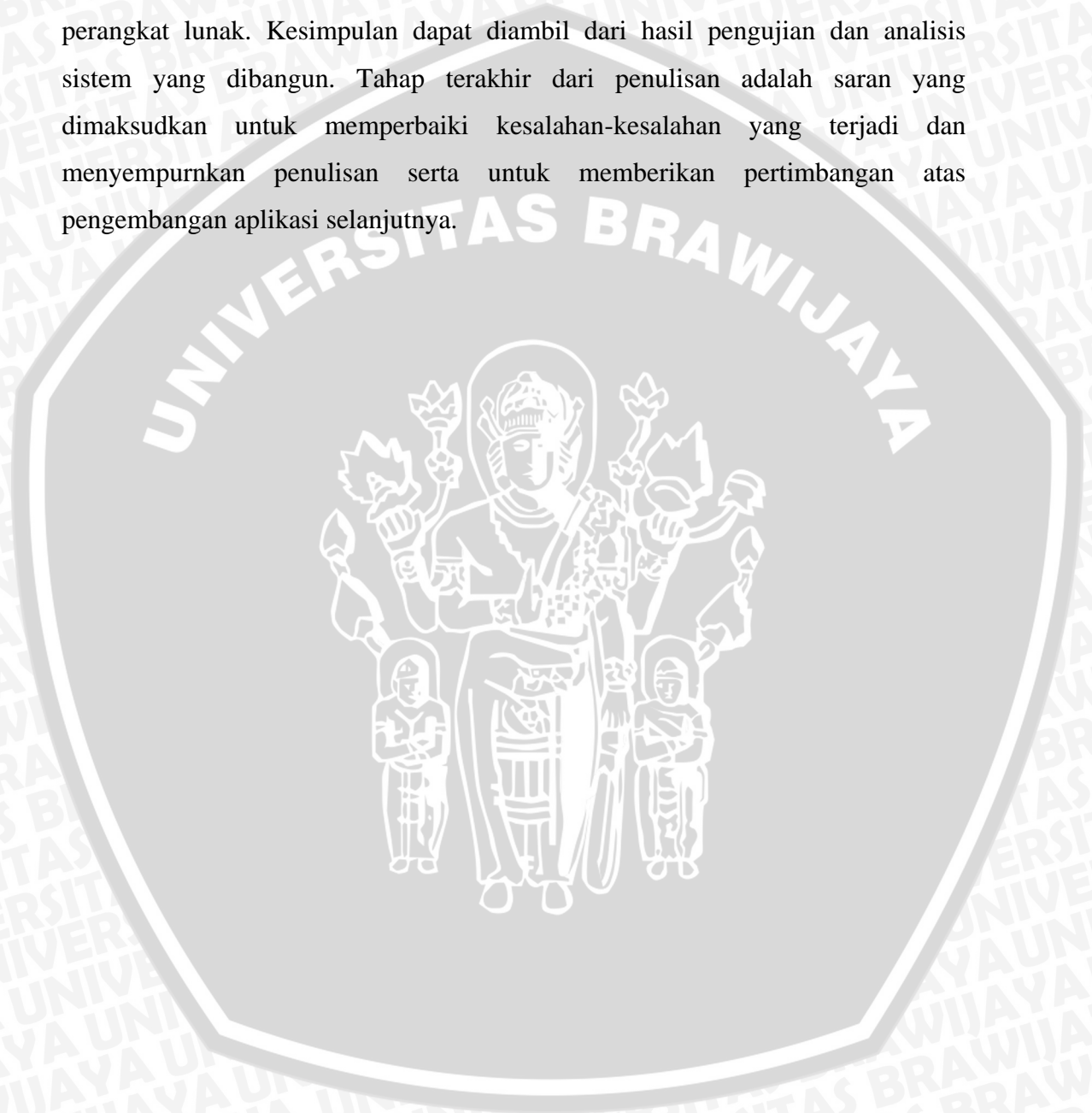
3.5 Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan untuk memastikan aplikasi yang dibangun sesuai dengan yang diinginkan. Pengujian perangkat lunak ini dilakukan dengan dua jenis pengujian. Pengujian perangkat lunak yang pertama dilakukan menggunakan metode *black-box*, yang berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya sehingga tidak terjadi kesalahan (*error*).

Pengujian perangkat lunak yang kedua dengan menggunakan pengujian non fungsional. Pengujian non fungsional merupakan pengujian yang bertujuan untuk mengetahui tingkat kenyamanan pengguna dalam mengoperasikan sistem. Pada pengujian perangkat lunak ini menggunakan kuisioner yang akan diisi oleh pengguna saat melakukan simulasi terhadap sistem perangkat lunak.

3.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dan saran dilakukan setelah selesai melakukan tahapan analisis kebutuhan, perancangan, implementasi, dan pengujian sistem perangkat lunak. Kesimpulan dapat diambil dari hasil pengujian dan analisis sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.



BAB IV

PERANCANGAN PERANGKAT LUNAK

Pada bab ini menjelaskan tentang analisis kebutuhan dan perancangan sistem. Proses perancangan perangkat lunak terdiri dari dua tahap. Tahap pertama adalah proses analisis kebutuhan kemudian dilanjutkan tahap kedua yaitu proses perancangan perangkat lunak. Tahap analisa kebutuhan terdiri dari dua langkah yaitu, identifikasi aktor dan identifikasi kebutuhan. Setelah tahap analisa kebutuhan dilakukan kemudian tahapan selanjutnya adalah proses perancangan. Pada proses perancangan sistem perangkat lunak memiliki lima tahap yaitu, perancangan *use case diagram*, perancangan kelas-kelas dan *interface* sehingga terhubung secara logika dengan *class diagram*, perancangan *activity diagram*, perancangan *sequence diagram*, dan yang terakhir adalah tahap perancangan tampilan antar muka pengguna sistem.

4.1 Analisis Kebutuhan

Pada analisa kebutuhan ini bertujuan untuk mendapatkan semua daftar kebutuhan yang dibutuhkan oleh sistem *quickcount* yang akan dibangun. Proses analisa kebutuhan ini diawali dengan proses identifikasi aktor yang terlibat dalam sistem dan kemudian dilanjutkan dengan penjelasan pada daftar kebutuhan ke dalam diagram *use case*.

4.1.1 Identifikasi Aktor

Pada tahap ini merupakan tahap awal dengan mengidentifikasi aktor-aktor yang berinteraksi dengan sistem *quickcount*.

Tabel 4. 1 Identifikasi Aktor

Aktor	Deskripsi
Administrator	Administrator adalah aktor yang mengoperasikan dan memelihara sistem <i>quickcount</i> . Dimana mempunyai hak akses untuk mengelola data, mengelola akun surveyor-surveyor baik mendaftarkan dan menghapus akun.
Surveyor	Surveyor adalah aktor yang memasukkan data hasil suara yang ada di TPS yang telah ditentukan oleh administrator.
Viewer	Viewer adalah aktor yang hanya bisa melihat hasil perolehan suara <i>quickcount</i> .

Pada Tabel 4.1 aktor yang berperan dalam sistem ini adalah *administrator*, *surveyor* dan *viewer*. Dalam hal ini *administrator* berperan dalam mengoperasikan dan memelihara sistem, *surveyor* berperan dalam mengolah hasil suara, dan *viewer* hanya berperan untuk melihat hasil dari sistem *quickcount*.

4.1.2 Identifikasi Kebutuhan

Dalam mengidentifikasi kebutuhan sistem *quickcount* mempunyai daftar kebutuhan berdasarkan kebutuhan fungsional yang ditunjukkan pada table 4.2.

Tabel 4. 2 Daftar Kebutuhan Fungsional

Nomor SRS	Kebutuhan	Aktor
SRS_001_01	Sistem dapat memasukan data kecamatan	Administrator
SRS_001_02	Sistem dapat menghapus data kecamatan	Administrator
SRS_001_03	Sistem dapat mengubah data kecamatan	Administrator
SRS_001_04	Sistem dapat memasukan data periode	Administrator
SRS_001_05	Sistem dapat menghapus data periode	Administrator
SRS_001_06	Sistem dapat mengubah data periode	Administrator
SRS_001_07	Sistem dapat memasukan data TPS	Administrator
SRS_001_08	Sistem dapat menghapus data TPS	Administrator
SRS_001_09	Sistem dapat mengubah data TPS	Administrator
SRS_001_10	Sistem dapat menambah data kandidat	Administrator
SRS_001_11	Sistem dapat menghapus data kandidat	Administrator
SRS_001_12	Sistem dapat mengubah data kandidat	Administrator
SRS_001_13	Sistem dapat menambah data user	Administrator
SRS_001_14	Sistem dapat menghapus data user	Administrator
SRS_001_15	Sistem dapat mengubah data user	Administrator
SRS_001_16	Sistem dapat menampilkan data <i>log quickcount</i>	Administrator
SRS_002_01	Sistem dapat memasukkan data hasil <i>quickcount</i>	Surveyor
SRS_002_02	Sistem dapat menghapus data hasil <i>quickcount</i>	Surveyor
SRS_002_03	Sistem dapat mengubah data hasil <i>quickcount</i>	Surveyor
SRS_003_01	Sistem dapat menampilkan hasil perolehan suara <i>quickcount</i>	Viewer

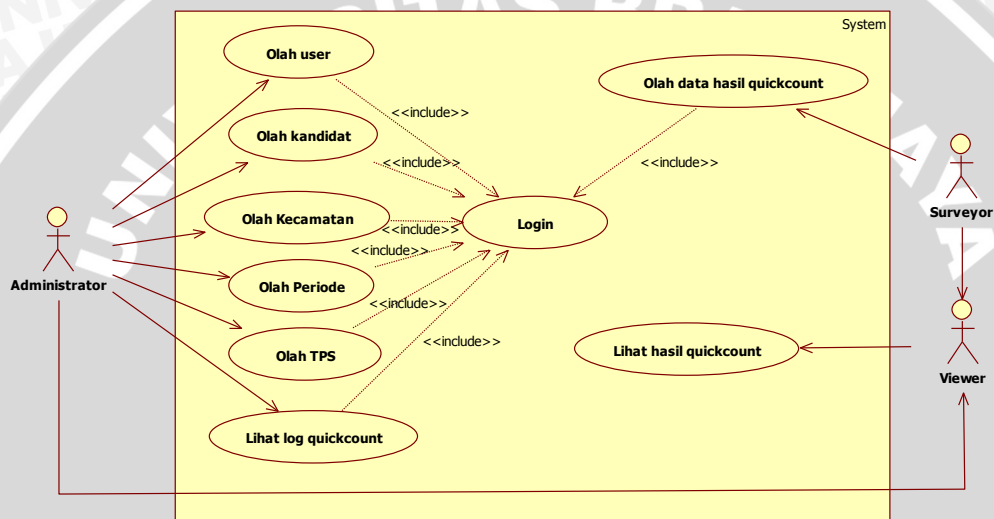
4.2 Perancangan Sistem Perangkat Lunak

Pada tahapan ini merupakan tahapan perancangan sistem perangkat lunak dari proses analisis yang telah dilakukakn sebelumnya. Perancangan ini dimodelkan dengan menggambarkan dalam bentuk diagram *use case*,

perancangan kelas-kelas dan *interface* sehingga terhubung secara logika dengan *class diagram*, perancangan *activity diagram*, perancangan *sequence diagram*, dan yang terakhir adalah tahap perancangan tampilan antar muka pengguna sistem.

4.2.1 Perancangan Use Case

Perancangan diagram *use case* ini melibatkan ketiga aktor yaitu administrator, surveyor dan viewer. Aktor yang berhubungan dengan *use case* merupakan penjelasan kondisi awal hingga akhir pada sistem quickcount.



Gambar 4. 1 Use Case Diagram

Penjelasan mengenai masing-masing *use case* pada Gambar 4.1 terangkum pada tabel-tabel dibawah ini:

1. *Use Case* Olah Data *Quickcount*

Tabel 4. 3 Use Case Olah Data Hasil Quickcount

Skenario Kasus pada Sistem	
Nama	Olah data hasil <i>quickcount</i>
Tujuan	<i>Surveyor</i> dapat melakukan olah data hasil <i>quickcount</i>
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana <i>surveyor</i> dapat melakukan olah data hasil <i>quickcount</i>
Aktor	<i>Surveyor</i>
Skenario Utama	
Kondisi awal	<i>Surveyor</i> telah <i>login</i> dan berada pada halaman utama <i>surveyor</i>
Aksi Aktor	Reaksi Sistem
- <i>Surveyor</i> menekan salah satu tautan <i>input / edit / delete</i> dihalaman <i>Surveyor</i>	- Sistem akan menampilkan halaman olah data hasil

<ul style="list-style-type: none"> - Jika <i>Surveyor</i> menekan tautan <i>input</i>, maka <i>surveyor</i> berada pada halaman olah data <i>input</i> dan memasukkan data hasil <i>quickcount</i> dengan menekan tombol <i>send</i> - Jika <i>Surveyor</i> menekan tautan <i>edit</i>, maka <i>surveyor</i> berada pada halaman olah data <i>edit</i>, kemudian dapat mengubah informasi data hasil <i>quickcount</i> dengan memilih salah satu data hasil dan menekan tombol <i>edit</i> - Jika <i>surveyor</i> ingin menghapus salah satu data hasil dengan menekan tombol <i>delete</i> 	<p><i>quickcount</i></p> <ul style="list-style-type: none"> - Sistem akan melakukan input data hasil <i>quickcount</i> dan menyimpannya - Sistem akan menampilkan data hasil <i>quickcount</i> yang ingin diubah dan kemudian meng-<i>update</i> data hasil <i>quickcount</i> yang telah diubah - Sistem akan menghapus data hasil <i>quickcount</i> yang telah dipilih untuk dihapus
<p>Kondisi Akhir</p>	<p>Sistem akan menyimpan data dan menampilkan data hasil <i>quickcount</i> yang sudah ditambah/dirubah ataupun dihapus</p>

Pada Tabel 4.3 *Use Case* Olah Data *Quickcount* merupakan skenario kasus pada sistem bekerja pada level *surveyor* yang menjelaskan bagaimana *surveyor* melakukan pengolahan data suara yang telah terkumpul. Dalam kasus ini *surveyor* telah *login* dan berada pada halaman utama *surveyor*. Pada halaman utama *surveyor* terdapat tautan untuk melakukan pengolahan data yaitu *input*, *edit*, dan *delete*. Jika *surveyor* memasukkan data hasil dengan menekan tombol input sehingga *surveyor* berada pada halaman olah data *input* dan memasukkan data hasil *quickcount* dengan menekan tombol *send* dan kemudian sistem akan menyimpannya. Jika *Surveyor* menekan tautan *edit*, maka sistem akan menampilkan halaman olah data *edit*, kemudian *surveyor* dapat mengubah informasi data hasil *quickcount* dengan memilih salah satu data hasil dan menekan tombol *edit*. Apabila sistem telah menampilkan form *edit* maka *surveyor* akan mengubah data dan tekan *save*. Jika *surveyor* ingin menghapus salah satu data hasil dengan menekan tombol *delete*.



2. Use Case Lihat Hasil Quickcount

Tabel 4. 4 Use Case Lihat Hasil Quickcount

Skenario Kasus pada Sistem	
Nama	Lihat hasil <i>quickcount</i>
Tujuan	<i>Administrator</i> dan <i>Viewer</i> dapat melihat hasil <i>quickcount</i>
Deskripsi	<i>Usecase</i> ini menjelaskan bagaimana <i>administrator</i> dan <i>viewer</i> dapat melihat hasil <i>quickcount</i>
Aktor	<i>Administrator</i> dan <i>Viewer</i>
Skenario Utama	
Kondisi awal	- <i>Administrator</i> telah <i>login</i> dan berada pada halaman utama <i>administrator</i> - <i>Viewer</i> berada pada halaman utama
Aksi Aktor	Reaksi Sistem
- <i>Administrator</i> dan <i>surveyor</i> menekan tautan lihat hasil <i>quickcount</i> dihalaman <i>administrator/surveyor</i> - <i>Viewer</i> berada pada halaman utama	Sistem akan menampilkan halaman lihat hasil <i>quickcount</i>
Kondisi Akhir	<i>Administrator</i> dan <i>Viewer</i> dapat melihat hasil <i>quickcount</i> yang ditampilkan oleh system

Pada Tabel 4.4 *use case* lihat hasil *quickcount* merupakan skenario kasus pada sistem yang bekerja pada *viewer*. Dalam hal ini *administrator* dan *surveyor* bisa mempunyai peran yaitu sebagai *viewer*. Dengan kondisi awal para aktor berada pada halaman utama, sistem telah memberikan hasil dari *quickcount* pada halaman utama tersebut. Jika aktor *administrator* dan *surveyor* telah *login* dan melakukan proses pengolahan data, maka *administrator* dan *surveyor* harus menekan tautan lihat hasil *quickcount* untuk melihat hasil *quickcount* yang terbaru.

Deskripsi perancangan *use case* selanjutnya dimulai dari *use case login*, *use case olah user*, *use case olah kandidat*, *use case kecamatan*, *use case periode*, *use case tps* dan *use case lihat log quickcount* terdapat pada lampiran 1 - lampiran 7.

4.2.2 Perancangan Activity Diagram

Activity diagram merupakan diagram yang menunjukkan kegiatan yang dilakukan oleh sistem yang akan dibangun oleh peneliti. *Activity diagram*

dikelompokkan berdasarkan sudut pandang aktor, yaitu *activity diagram login*, *activity diagram olah user*, *activity diagram olah kandidat*, *activity diagram olah data hasil quickcount*, *activity diagram lihat log quickcount*, dan *activity diagram lihat hasil quickcount*.

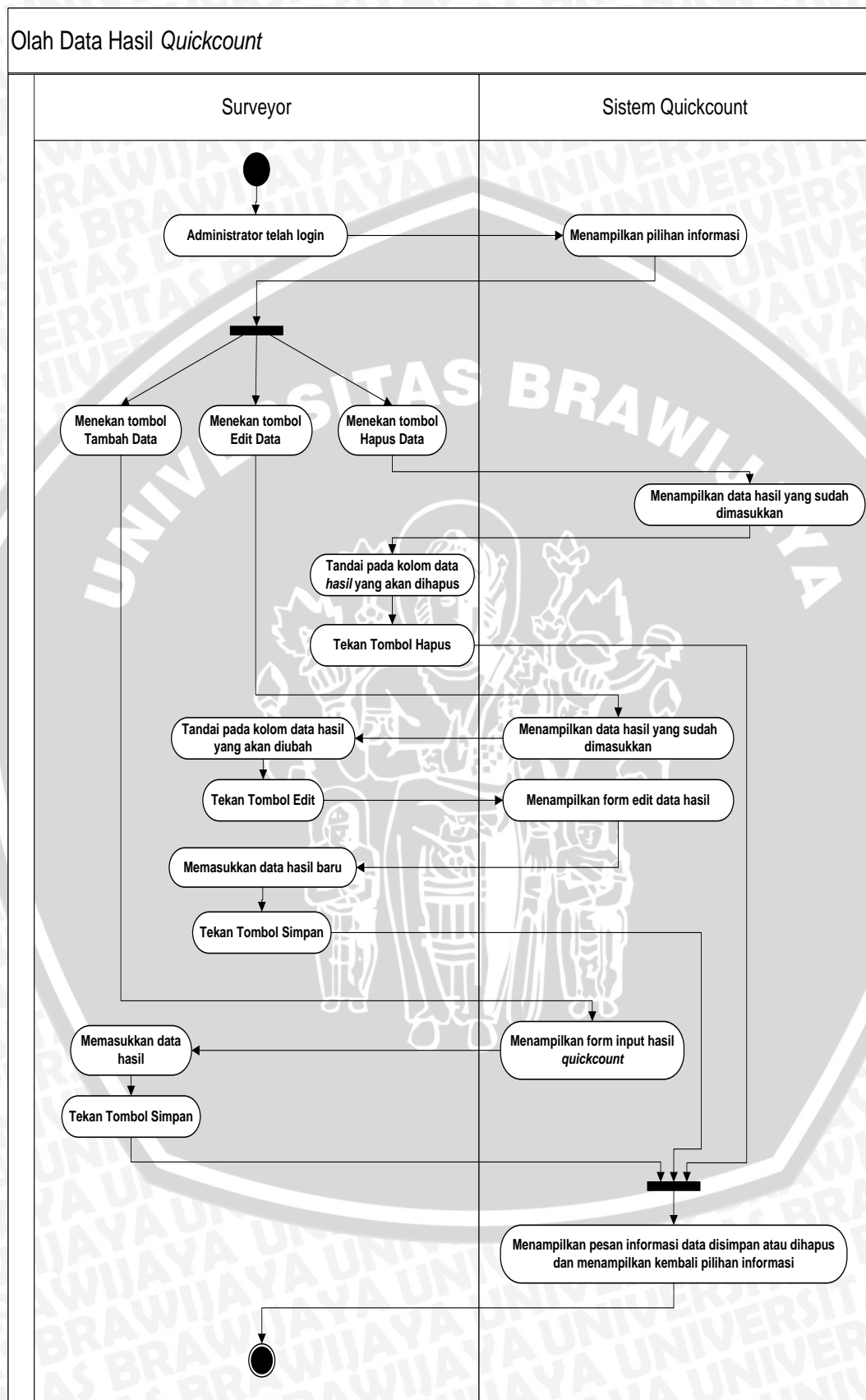
1. *Activity diagram* olah data hasil *quickcount*

Pada Gambar 4.2 *Activity diagram* olah data hasil *quickcount* merupakan gambaran aktifitas yang dilakukan oleh *surveyor* untuk melakukan pengolahan data hasil *quickcount*. Pada aktifitas ini *surveyor* sudah berhasil *login* dan sistem menampilkan pilihan informasi.

Jika akan memasukkan data hasil *surveyor* menekan tombol tambah data dan sistem akan menampilkan form input hasil *quickcount*. *Surveyor* akan mengisi form input dengan hasil yang diperoleh dari tempat pemungutan suara dan kemudian menekan tombol simpan sehingga sistem dapat menyimpan. Apabila proses pengisian berhasil maka sistem akan menampilkan pesan informasi hasil data *quickcount* telah tersimpan.

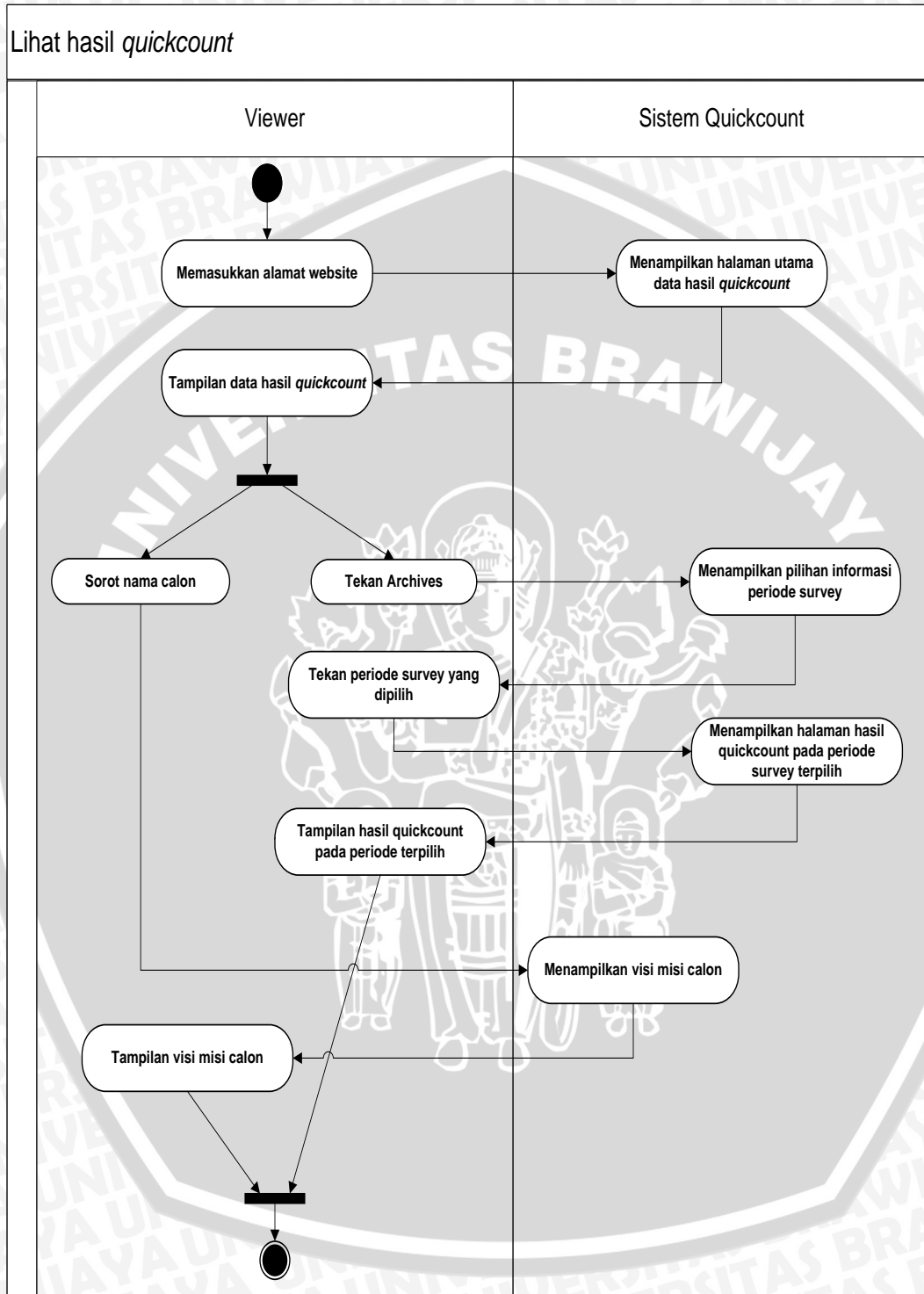
Jika akan mengubah data hasil *surveyor* menekan tombol edit data dan sistem akan menampilkan data hasil yang sudah dimasukkan dalam database. *Surveyor* akan menandai kolom data hasil yang akan diubah dan kemudian menekan tombol edit dan sistem akan menampilkan form edit data hasil *quickcount*. *Surveyor* akan mengisi form edit data dengan hasil terbaru yang diperoleh dari tempat pemungutan suara dan kemudian menekan tombol simpan sehingga sistem dapat menyimpan. Apabila proses pengubahan berhasil maka sistem akan menampilkan pesan informasi hasil data *quickcount* telah tersimpan.

Jika *surveyor* akan menghapus maka menekan tombol hapus data dan sistem akan menampilkan data hasil yang sudah dimasukkan. *Surveyor* memilih dengan menandai pada kolom data hasil yang akan dihapus dan menekan tombol hapus. Apabila proses menghapus data hasil berhasil maka sistem akan menampilkan pesan informasi hasil data *quickcount* terpilih telah dihapus.



Gambar 4. 2 Activity diagram olah data hasil quickcount

2. Activity diagram lihat hasil quickcount



Gambar 4. 3 Activity Diagram lihat hasil quickcount

Pada Gambar 4.3 activity diagram lihat hasil quickcount merupakan skenario kasus pada sistem yang bekerja pada viewer. Dalam hal ini administrator dan surveyor bisa mempunyai peran yaitu sebagai viewer. Dengan kondisi awal

para aktor memasukkan alamat website dan sistem menampilkan halaman utama yang berisikan data hasil *quickcount*. Jika aktor *administrator* dan *surveyor* telah *login* dan melakukan proses pengolahan data, maka *administrator* dan *surveyor* harus menekan tautan lihat hasil *quickcount* untuk melihat hasil *quickcount* yang terbaru. Didalam halaman data hasil *quickcount* menyediakan pilihan informasi yaitu visi misi calon dan *archive quickcount* yang pernah dilakukan.

Jika *viewer* menekan foto calon maka sistem akan menampilkan visi misi calon. Dan jika *viewer* menekan tombol *archives* maka sistem akan menampilkan pilihan informasi periode *survey*. Kemudian *viewer* dapat menekan tautan periode *survey* yang dipilih sehingga sistem dapat menampilkan hasil *quickcount* pada periode terpilih.

Perancangan *activity diagram* selanjutnya dimulai dari *activity diagram login*, *activity diagram* olah user, *activity diagram* olah kandidat, *activity diagram* kecamatan, *activity diagram* periode, *activity diagram* tps dan *activity diagram* lihat log *quickcount* terdapat pada lampiran 8 - lampiran 14.

4.2.3 Perancangan *Sequence Diagram*

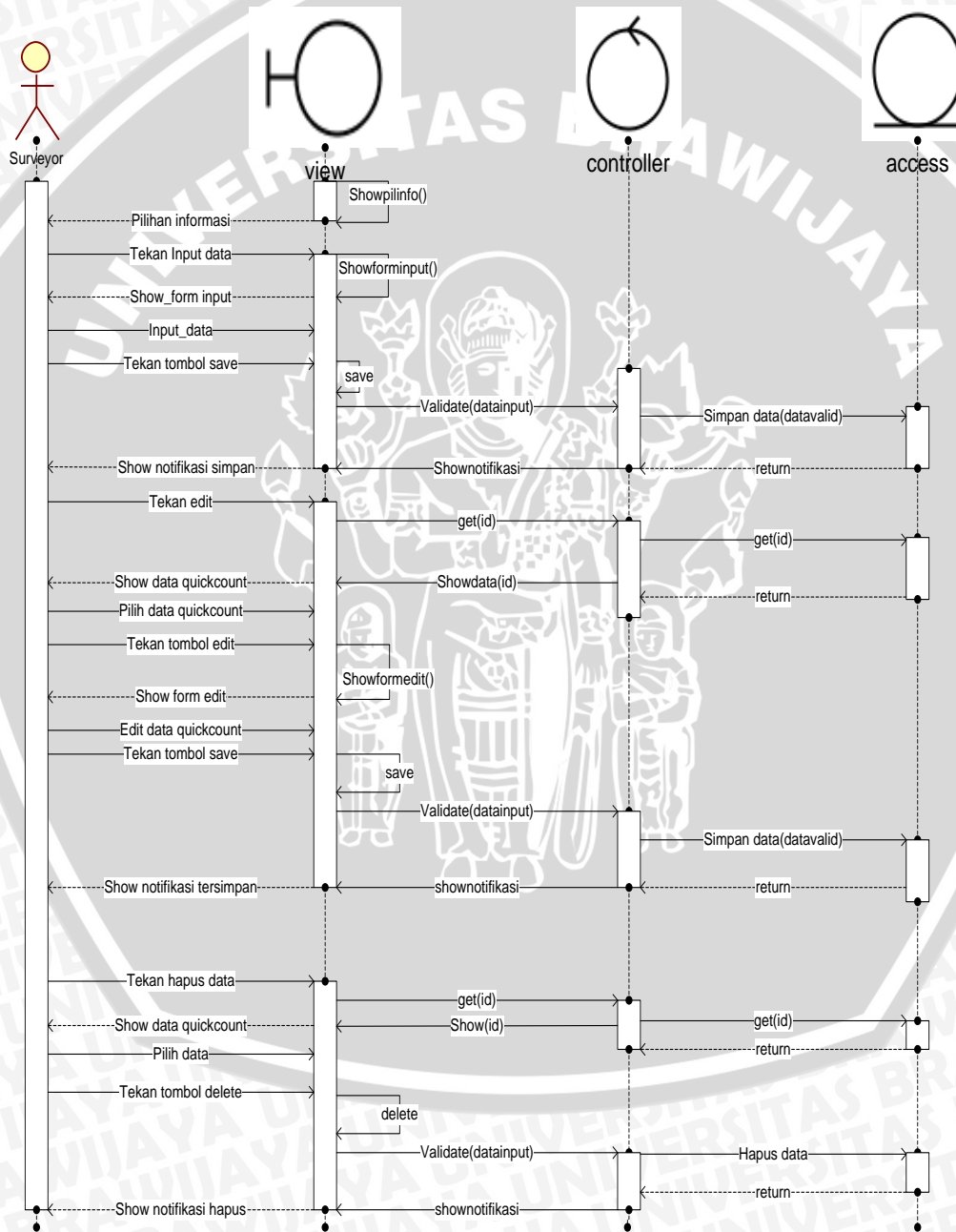
Sequence diagram merupakan perancangan interaksi untuk menjelaskan interaksi antar objek pada sistem. Dalam hal ini *sequence diagram* digunakan untuk menggambarkan skenario yang akan terjadi pada sistem dan akan mendapatkan keluaran tertentu. Berikut merupakan skenario yang digambarkan dengan *sequence diagram*:

1. *Sequence diagram* olah data hasil

Pada Gambar 4.4 *Sequence diagram* olah data hasil *quickcount* merupakan gambaran interaksi yang dilakukan oleh *surveyor* dengan sistem agar dapat mengolah data hasil suara *quickcount*. Pada interaksi ini *surveyor* sudah melakukan *login* dan sistem menampilkan pilihan informasi menu olah data *quickcount*.

Jika *surveyor* akan memasukkan data perolehan suara, maka *surveyor* mengirimkan pesan input pada sistem *view* dan kemudian sistem *view* menampilkan *form input*. Ketika *form input* telah ditampilkan, *surveyor* akan melakukan *input* data dan menekan tombol *save* pada sistem *view*. Sistem *view*

akan menerima dan mengirim pesan *validate* pada sistem *controller*. Jika sistem *controller* telah melakukan *validate (data input)* maka sistem *controller* akan mengirimkan pesan simpan data pada sistem *access*. Sistem *access* mengirimkan *return* pada sistem *controller* dan selanjutnya sistem *controller* mengirimkan *notifikasi* pada sistem *view*. Sistem *view* akan melanjutkan pesan *notifikasi* tersebut pada tampilan yang akan diterima oleh *surveyor*.



Gambar 4. 4 Sequence diagram olah data hasil quickcount

Jika surveyor akan melakukan perubahan data maka menekan *edit* sehingga sistem *view* akan mengirimkan pesan *get* data pada sistem *controller* kemudian sistem *controller* melanjutkan ke sistem *access*. Sistem *access* akan mengirimkan pesan *return* pada sistem *controller* dan sistem *controller* mengirim pesan *show* data quickcount pada sistem *view*, sehingga *surveyor* dapat mengetahui data yang telah tersimpan pada sistem *quickcount*. *Surveyor* akan memilih data pada sistem *view* dan kemudian sistem *view* mengirimkan kembali form *edit* data. Apabila *surveyor* telah mengubah data *quickcount* dan menekan tombol *save* maka pesan *edit* akan dikirimkan pada sistem *view* dan sistem *view* akan mengirimkan pesan *validate* ke sistem *controller*. Jika telah *valid* sistem *controller* akan mengirimkan pesan simpan data pada sistem *access* dan memberikan *notifikasi* pada sistem *view*. Sistem *view* akan melanjutkan pesan *notifikasi* tersebut pada tampilan yang akan diterima oleh *surveyor*.

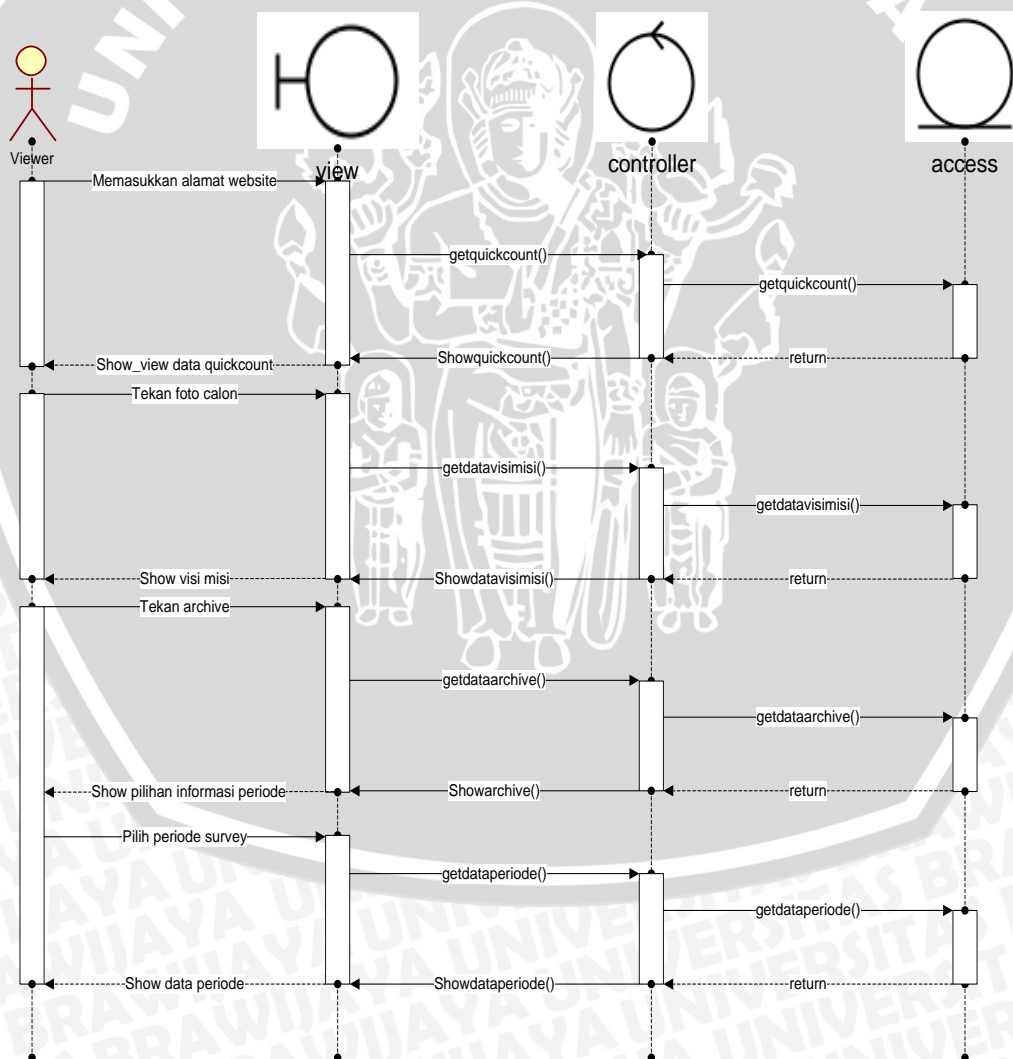
Jika *surveyor* mengirimkan pesan hapus data pada sistem, maka sistem *view* akan mengirimkan pesan *get* data pada sistem *controller* yang dilanjutkan ke sistem *access*. Setelah sistem *access* menerima pesan informasi *get* data, maka sistem *access* akan mengirimkan pesan *return* pada sistem *controller* yang akan dilanjutkan oleh sistem *controller* dengan mengirimkan pesan berupa data *quickcount* yang tersimpan ke sistem *view*. Sistem *view* akan menampilkan data *quickcount* pada *surveyor* sehingga *surveyor* dapat menandai data yang akan dihapus. Apabila proses penandaan selesai, *surveyor* dapat mengirimkan pesan hapus data *quickcount* yang terpilih pada sistem *view* dan menekan tombol *delete*. Sistem *view* mengirimkan pesan validasi pada sistem *controller* yang akan dilanjutkan oleh sistem *controller* untuk mengirim pesan hapus data pada *access*. Kemudian sistem *controller* mengirimkan pesan notifikasi bahwa data telah dihapus pada sistem *view* dan sistem *view* mengirimkan notifikasi pada *surveyor*.

2. Sequence diagram Lihat Hasil Quickcount

Pada Gambar 4.5 *Sequence Diagram* lihat hasil *quickcount* merupakan gambaran interaksi pada proses lihat hasil *quickcount*. Pada interaksi ini dilakukan oleh *viewer* terhadap sistem *quickcount*. Dalam hal ini *viewer* tanpa melakukan login dapat melihat hasil perolehan suara *quickcount*. *Viewer* memasukkan alamat website pada sistem *view* yang kemudian dilanjutkan sistem *view* mengirimkan

pesan *get* data pada sistem *controller*. Sistem *controller* melanjutkan pesan tersebut pada *access* yang kemudian sistem *access* menerima dan mengirim pesan *return* pada sistem *controller*. Setelah sistem *controller* menerima pesan tersebut, sistem *controller* mengirimkan data hasil *quickcount* pada sistem *view*. Dalam hal ini *viewer* bisa melihat hasil *quickcount* tersebut.

Apabila *viewer* ingin mengetahui lebih jelas visi misi calon maka *viewer* dapat menekan foto calon pada sistem *view* yang kemudian dilanjutkan sistem *view* mengirimkan pesan *get* data pada sistem *controller*. Sistem *controller* melanjutkan pesan tersebut pada *access* yang kemudian sistem *access* menerima dan mengirim pesan *return* pada sistem *controller* dan dilanjutkan sistem *controller* mengirimkan data visi misi calon pada sistem *view*.



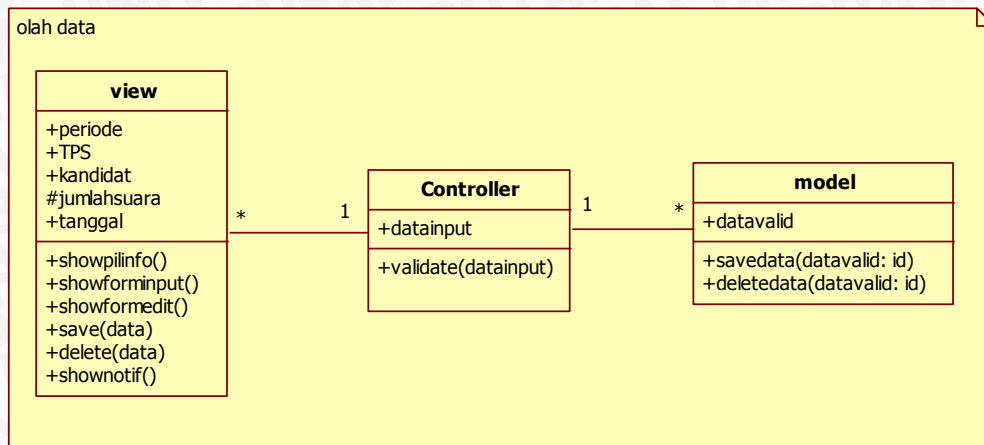
Gambar 4. 5 Sequence diagram Lihat Hasil Quickcount

Apabila *viewer* ingin mengetahui tentang periode *quickcount* yang telah dilakukan, maka *viewer* dapat menekan *archive* pada tampilan sistem *view* yang kemudian dilanjutkan sistem *view* mengirimkan pesan *get* data pada sistem *controller*. Sistem *controller* melanjutkan pesan tersebut pada *access* yang kemudian sistem *access* menerima dan mengirim kembali pesan *return* pada sistem *controller*. Sistem *controller* menerima dan mengirimkan pilihan informasi periode *survey* pada sistem *view*. *Viewer* bisa memilih periode *survey quickcount* yang ingin dilihat dengan menekan salah satu periode *survey* pada sistem *view* dan kemudian dilanjutkan sistem *view* mengirimkan pesan *get* data pada sistem *controller*. Sistem *controller* melanjutkan pesan tersebut pada *access* yang kemudian sistem *access* menerima dan mengirim kembali pesan *return* pada sistem *controller*. Sistem *controller* mengirimkan hasil *quickcount* pada periode terpilih pada sistem *view*. Sehingga *viewer* dapat melihat hasil *survey* di tiap-tiap periode *survey*.

Perancangan *sequence diagram* selanjutnya dimulai dari *sequence diagram login*, *sequence diagram* olah data *surveyor*, *sequence diagram* olah kandidat, *sequence diagram* kecamatan, *sequence diagram* periode, *sequence diagram* tps, dan *sequence diagram* lihat *log quickcount* terdapat pada lampiran 15 - lampiran 21.

4.2.4 Perancangan Class Diagram

Perancangan *class diagram* diperlukan untuk membantu dalam menggambarkan keadaan struktur dan deskripsi *class*, *package* dan objek pada sistem yang dirancang. Pada Gambar 4.6 perancangan *class diagram* olah data hasil *quickcount* merupakan gambaran beberapa kelas-kelas yang dapat menunjukkan relasi-relasi pada sistem *quickcount*.



Gambar 4. 6 Perancangan Class Diagram Olah Data Hasil Quickcount

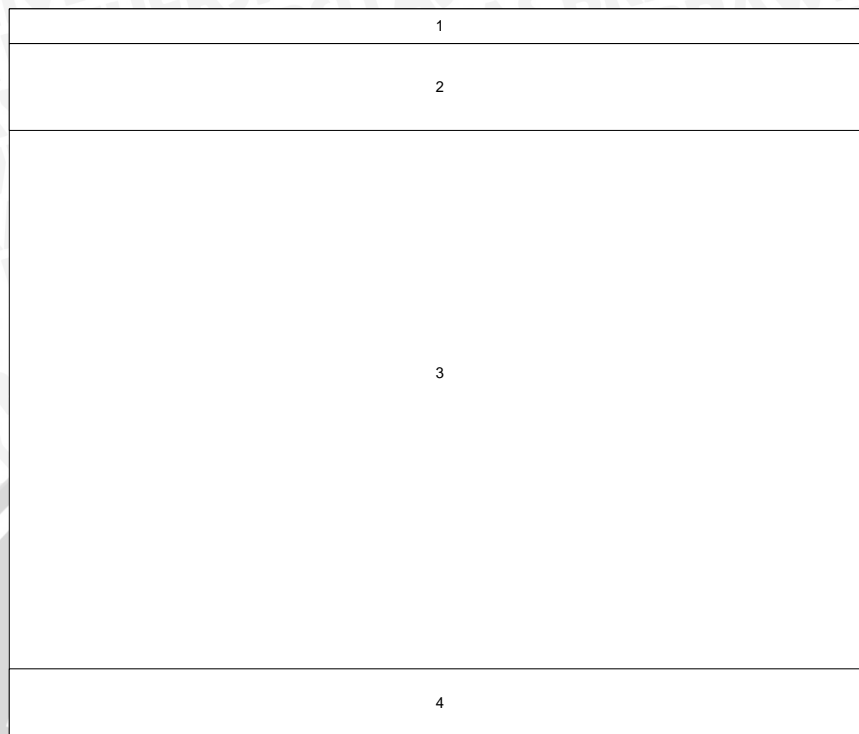
4.2.5 Perancangan Antarmuka

Perancangan antarmuka diperlukan untuk membantu merancang tampilan sistem yang akan dibuat sehingga dapat mewakili keadaan sebenarnya dari sistem yang akan dibangun. Dalam hal ini perancangan antarmuka dibuat dari perancangan halaman utama sampai perancangan halaman untuk viewer.

1. Perancangan Halaman Utama / Viewer

Halaman utama digunakan untuk menampilkan hasil *quickcount* bagi *viewer* dan untuk proses *login* bagi *surveyor* dan *administrator*. Gambar 4.7 memiliki keterangan sebagai berikut:

1. Header yang berisi *tool bar* pilihan halaman administrasi dan informasi sistem
2. Content Heder berisi identitas sistem *quickcount*
3. Content yang berisi hasil *quickcount*
4. Footer berisi nama dan tahun dari sistem



Gambar 4. 7 Perancangan Antarmuka halaman utama/viewer

2. Perancangan Halaman Input Hasil oleh *Surveyor*

Halaman input hasil oleh *suveyor* digunakan untuk menampilkan konten apa saja yang dapat dikelola oleh *suveyor*. Gambar 4.8 memiliki keterangan sebagai berikut:

1. Header yang berisi logo dan identitas sistem *quickcount*
2. Berisikan *form logout* bagi *surveyor*
3. Menu navigasi
4. Menu navigasi khusus
5. Konten yang berisi pilihan pengolahan apa yang akan dilakukan *administrator* selanjutnya
6. Footer berisi nama dan tahun dari sistem

Surveyor	1 2 <input type="button" value="Logout"/>
3	4 periode <input type="text"/> TPS <input type="text"/> Kandidat <input type="text"/> Jumlah suara <input type="text"/> Tanggal input <input type="text"/> <input type="button" value="save"/> 5
6	

Gambar 4. 8 Perancangan Antarmuka halaman input hasil oleh surveyor

Perancangan antarmuka selanjutnya dimulai dari perancangan halaman olah data oleh *administrator*, perancangan halaman *surveyor* dan perancangan halaman olah data oleh *surveyor* terdapat pada lampiran 16 - lampiran 18.

BAB V IMPLEMENTASI PERANGKAT LUNAK

Bab ini membahas mengenai tahapan-tahapan implementasi sistem *quickcount* berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak. Implementasi perangkat lunak terdiri dari spesifikasi lingkungan sistem, batasan-batasan implementasi sistem, implementasi basis data, implementasi *file* program, dan implementasi antarmuka.

5.1 Spesifikasi Lingkungan Sistem

Sistem *quickcount* dibangun dalam lingkungan perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Lingkungan Perangkat Keras

Spesifikasi lingkungan perangkat keras merupakan suatu perangkat yang digunakan dalam pengembangan sistem *quickcount* yang dijelaskan pada Tabel 5.1 dibawah ini

Tabel 5. 1 Spesifikasi Lingkungan Perangkat Keras

Acer Aspire 4750	
Processor	Intel (R) Core (TM) i3-2310M CPU @ 2.10GHz
RAM	2 GB
Hardisk	500 GB
Graphic card	Intel(R) HD Graphics 300

5.1.2 Spesifikasi Lingkungan Perangkat Lunak

Spesifikasi lingkungan perangkat lunak merupakan suatu perangkat lunak yang digunakan dalam pengembangan sistem *quickcount* yang dijelaskan pada Tabel 5.2 dibawah ini

Tabel 5. 2 Spesifikasi Lingkungan Perangkat Lunak

Acer Aspire 4750	
<i>Operating system</i>	Windows 7 Ultimate 32-bit
<i>Programming Language</i>	<i>HyperText Markup Language (HTML), HyperText Preprocessor (PHP)</i>
<i>Software Development Kit</i>	Google Chrome
<i>Basis Data Management System</i>	MySQL
<i>Intergated Development Environment</i>	Portable Dreamweaver 8.0.2

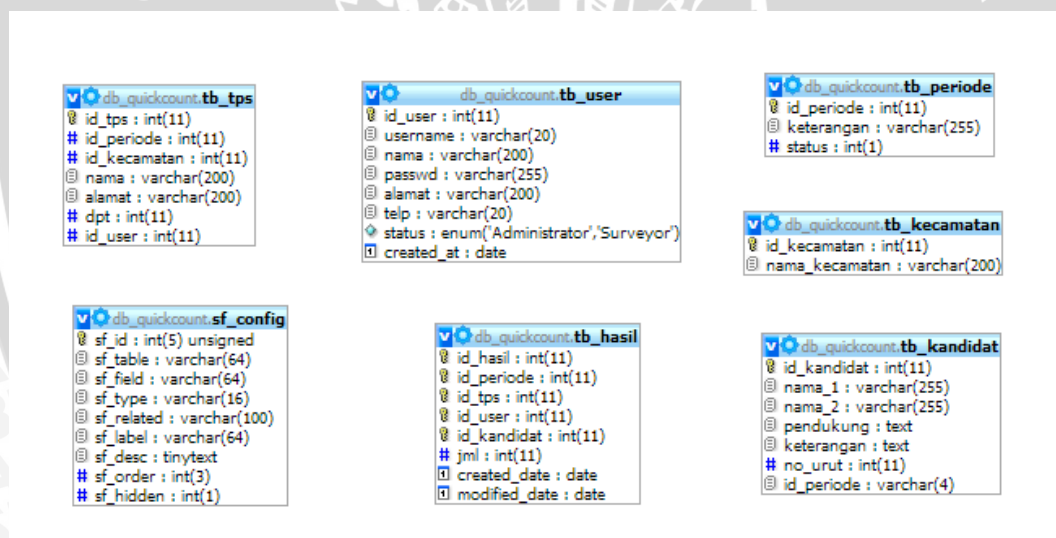
5.2 Batasan Implementasi Sistem

Batasan dalam mengimplementasikan sistem *quickcount* adalah sebagai berikut:

1. Perangkat lunak ini dibangun dengan menggunakan bahasa pemrograman PHP dengan *Framework CodeIgniter*
2. *Database Management System* yang digunakan adalah MySQL
3. Server yang digunakan dalam pengembangan aplikasi adalah Apache Server
4. Perangkat lunak ini berbasis web yang dapat dijalankan pada browser

5.3 Implementasi Basis Data

Implementasi penyimpanan data dilakukan dengan menggunakan basis data *management system* MySQL. Gambar 5.1 menunjukkan basis data yang dilakukan oleh sistem *quickcount*.



Gambar 5. 1 Basis Data Sistem Quickcount

Pada Gambar 5.1 merupakan implementasi basis data yang digunakan pada sistem *quickcount*. Implementasi basis data mempunyai beberapa tabel, yaitu tabel *tb_tps* sebagai basis data untuk data TPS, tabel *tb_user* sebagai basis data untuk data user, *tb_config*, *tb_hasil* sebagai basis data untuk data hasil suara, tabel *tb_periode*, tabel *tb_kecamatan* sebagai basis data untuk data kecamatan, dan tabel *tb_kandidat* sebagai basis data untuk data kandidat yang dipilih.

5.4 Implementasi *File Program*

Implementasi *file* program merupakan tahapan implementasi pengkodean program yang diterapkan sesuai konsep *Model, View, dan Controller*.

Controller `tb_hasil.php`

```
function validateData($val) {  
    $this->load->library('form_validation');  
    $val->form_validation->set_rules( 'jml', lang('jml'), '11' );  
    $val->form_validation->set_rules( 'id_periode',  
    lang('id_periode'), '11' );  
    $val->form_validation->set_rules( 'id_tps', lang('id_tps'),  
    '11' );  
    $val->form_validation->set_rules( 'id_kandidat',  
    lang('id_kandidat'), '11' );  
    return $val->form_validation->run();  
}
```

Gambar 5. 2 Implementasi file program Controller `tb_hasil.php`

Pada Gambar 5.2 merupakan implementasi *file* program controller `tb_hasil.php` dari perancangan sistem *quickcount*. Dalam hal ini data input yang ditunjuk pada Gambar 4.6 yang dimasukkan dalam controller merupakan *array* yang berupa data jumlah suara, periode, tps, kandidat, dan tanggal sehingga dapat melakukan validasi dalam pengolahan data pada sistem *quickcount*.

Models `tb_hasil.php`

```
function insert ( $data )  
{  
    $this->db->insert( 'tb_hasil', $data );  
    return $this->db->insert_id();  
}  
function update ( $id, $data )  
{  
    $this->db->where( 'tb_hasil.id_kandidat', $id );  
    $this->db->update( 'tb_hasil', $data );  
}  
function delete ( $id )  
{
```



```

if( is_array( $id ) )
{
    $this->db->where_in( 'tb_hasil.id_kandidat', $id );
}
else
{
    $this->db->where( 'tb_hasil.id_kandidat', $id );
}
$this->db->delete( 'tb_hasil' );
}

```

Gambar 5. 3 Implementasi file program Models_hasil.php

Pada Gambar 5.3 implementasi file program models_hasil.php dari perancangan yang telah buat pada bab sebelumnya. Dalam hal ini menyimpan data dilakukan dengan dua cara, yaitu menyimpan data pertama dengan cara memasukkan data (*insert data*) dan menyimpan dengan cara memperbarui data (*update data*). Sehingga dalam pengkodean *file program* save_data dilakukan dengan cara insert_data dan update_data.

List_tb_hasil.tpl

```

<tbody>
{foreach $tb_hasil_data as $row}
  <tr class="{cycle values='odd,even'}">
    <td><input type="checkbox" class="checkbox" name="delete_ids[]"
value="{ $row.id_kandidat}" /></td>
    <td>{ $row.id_kandidat}</td>
    <td>{ $row.jml}</td>
    <td>{ $row.id_periode}</td>
    <td>{ $row.id_tps}</td>
    <td>{ $row.created_date}</td>

    <td width="80">
      <a href="tb_hasil/show/{ $row.id_kandidat}"></a>
      <a href="tb_hasil/edit/{ $row.id_kandidat}"></a>
      <a href="javascript:chk('tb_hasil/delete/{ $row.id_kandidat}')"></a>
    </td>
  </tr>
{/foreach}
</tbody>

```

Gambar 5. 4 Implementasi file program List_tb_hasil.tpl

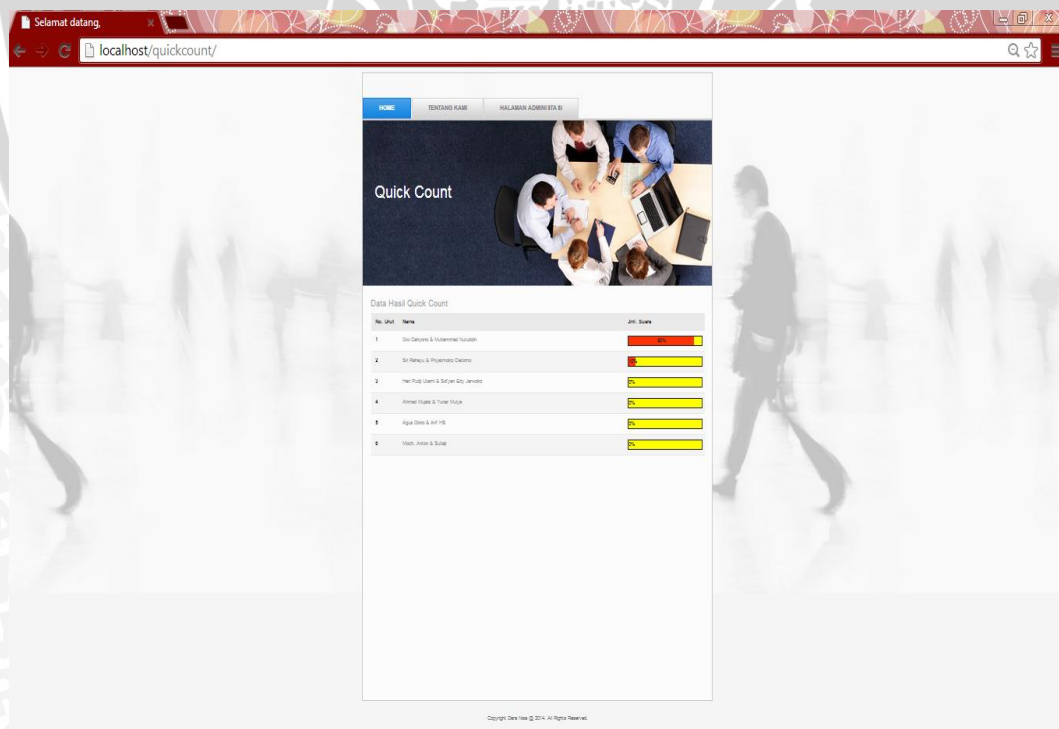
Pada Gambar 5.4 Implementasi *file* program *list_tb_hasil.tpl* merupakan implementasi *view* tabel hasil dari perancangan sistem *quickcount*. Dalam hal ini attribute yang akan dimasukkan dalam pengolahan adalah periode, TPS, kandidat, jumlah suara, dan tanggal penginputan data.

5.5 Implementasi Antarmuka

Implementasi antarmuka dilakukan untuk mempermudah pengguna dalam berinteraksi dengan sistem *quickcount*. Antarmuka sistem *quickcount* ini dibagi menjadi 3 bagian yaitu antarmuka bagi viewer, antarmuka bagi administrator, dan antarmuka bagi surveyor.

5.5.1 Antarmuka bagi Viewer

Antarmuka bagi *viewer* merupakan halaman utama saat pertama kali pengguna menjalankan sistem *quickcount* ini. Dalam hal ini *administrator* dan *surveyor* juga berperan sebagai *viewer*. Sesuai perancangan antarmuka pada Gambar 4.7, implementasi antarmuka dapat dilakukan dengan memiliki keterangan sebagai berikut

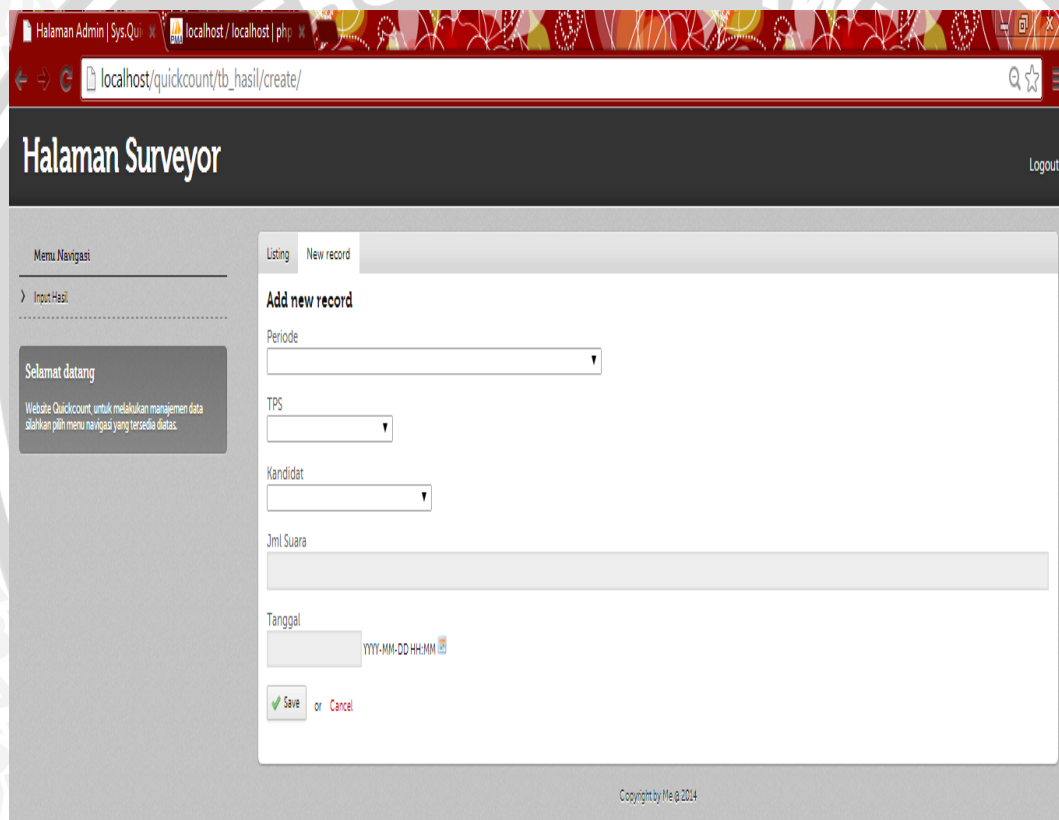


Gambar 5. 5 Imlementasi Antarmuka bagi Viewer

Pada Gambar 5.5 merupakan implementasi antarmuka bagi *viewer* yang menampilkan perolehan suara dalam bentuk prosentase hasil suara *quickcount*. Dalam antarmuka *viewer* ini juga menampilkan pilihan navigasi untuk masuk ke halaman administrasi.

5.5.2 Antarmuka bagi *Surveyor*

Antarmuka bagi *surveyor* merupakan beberapa implementasi antarmuka saat pertama kali *surveyor* melakukan *login*, halaman utama *surveyor* dan mengolah data hasil *quickcount*. Sesuai perancangan antarmuka pada Gambar 4.8, implementasi antarmuka input data hasil oleh *surveyor*



The screenshot displays a web browser window with the URL `localhost/quickcount/tb_hasil/create/`. The page title is "Halaman Surveyor" and includes a "Logout" link. On the left, there is a navigation menu with "Input Hasil" selected and a "Selamat datang" message. The main content area is titled "Add new record" and contains the following form fields:

- Periode: A dropdown menu.
- TPS: A dropdown menu.
- Kandidat: A dropdown menu.
- Jml Suara: A text input field.
- Tanggal: A date picker showing "YYYY-MM-DD HH:MM".

At the bottom of the form are "Save" and "Cancel" buttons. A copyright notice "Copyright © Me @ 2014" is visible at the bottom of the page.

Gambar 5. 6 Implementasi Antarmuka Input Data oleh Surveyor

Pada Gambar 5.6 implementasi antarmuka input data oleh *surveyor* merupakan halaman untuk memasukkan data hasil *quickcount*. Dalam memasukkan data hasil *quickcount* dibutuhkan data periode survey, data TPS yang disurvei, kandidat dan jumlah hasil suara.

BAB VI

PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian dan analisis sistem *quickcount*. Proses pengujian sistem dilakukan dengan menggunakan pengujian validasi dan proses analisis dilakukan sesuai dengan hasil pengujian validasi.

6.1 Pengujian

Proses pengujian sistem dilakukan dengan menggunakan pengujian validasi dan pengujian non fungsional. Pengujian bertujuan untuk mengetahui beroperasinya sistem agar dapat memenuhi kebutuhan pengguna.

6.1.1 Pengujian Validasi

Pengujian validasi dilakukan untuk mengetahui apakah sistem *quickcount* sudah sesuai dengan daftar kebutuhan yang telah ditentukan sebelumnya. Daftar kebutuhan yang telah dilakukan di bab sebelumnya akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode *Blackbox-Testing* yang berguna untuk mengetahui beroperasinya sistem *quickcount* dan kesesuaian dengan perancangannya.

6.1.1.1. Kasus Uji Validasi Penambahan Data Hasil *Quickcount*

Tabel 6. 1 Kasus Uji Validasi Penambahan Data Hasil *Quickcount*

Nama Kasus Uji	Penambahan Data Hasil <i>Quickcount</i>
Tujuan Pengujian	Untuk menambahkan data hasil <i>quickcount</i> oleh <i>surveyor</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Surveyor</i> melakukan login dengan mengisi kolom <i>username</i> dengan kata server, kolom <i>password</i> dengan kata server, dan menekan tombol <i>login</i> 2. <i>Surveyor</i> masuk ke halaman input hasil dengan menekan tombol input hasil 3. <i>Surveyor</i> masuk ke halaman <i>new record</i> dengan menekan tombol <i>new record</i>

	<p>4. <i>Surveyor</i> memasukkan data pada form <i>input</i> yang disediakan yang berupa periode yang diisi dengan periode 2013-2018, TPS diisi blimbing1 TPS 1, Kandidat diisi dengan pasangan dwi-udin, jumlah suara 12</p> <p>5. <i>Surveyor</i> menyimpan data hasil <i>quickcount</i> dengan menekan tombol <i>save</i></p>
Hasil yang diharapkan	Sistem dapat menyimpan data hasil <i>quickcount</i> baru. Masuk ke halaman <i>input</i> hasil dan melihat data hasil <i>quickcount</i> baru yang terisi.
Hasil yang didapatkan	Sistem menyimpan dan menampilkan data hasil <i>quickcount</i> pada TPS blimbing1 TPS1 pasangan nomor urut 1 dalam data hasil
Status Validitas	Valid

Pada Tabel 6.1 merupakan kasus uji validasi penambahan data hasil *quickcount* yang bertujuan untuk menambahkan data hasil *quickcount* oleh *surveyor*. Dalam hal ini prosedur uji yang dilakukan ketika *surveyor* melakukan login dengan mengisi kolom *username* dengan kata server, kolom *password* dengan kata server, dan menekan tombol *login*. Kemudian *Surveyor* masuk ke halaman input hasil dengan menekan tombol input hasil. *Surveyor* masuk ke halaman *new record* dengan menekan tombol *new record*. *Surveyor* memasukkan data pada form *input* yang disediakan yang berupa periode yang diisi dengan periode 2013-2018, TPS diisi blimbing1 TPS 1, Kandidat diisi dengan pasangan dwi-udin, jumlah suara 12. *Surveyor* menyimpan data hasil *quickcount* dengan menekan tombol *save*. Hasil yang diharapkan dari prosedur uji agar sistem dapat menyimpan data hasil *quickcount* baru. Masuk ke halaman *input* hasil dan melihat data hasil *quickcount* baru yang terisi. Hasil yang didapatkan sistem menyimpan dan menampilkan data hasil *quickcount* pada TPS blimbing1 TPS1 pasangan nomor urut 1 dalam data hasil. Sehingga kasus uji validasi penambahan hasil *quickcount* telah valid

6.1.1.2. Kasus Uji Validasi Penambahan Data Hasil Quickcount Non Ideal

Tabel 6. 2 Kasus Uji Validasi Penambahan Data Hasil Quickcount Non Ideal

Nama Kasus Uji	Penambahan Data Hasil <i>Quickcount</i>
Tujuan Pengujian	Untuk menambahkan data hasil <i>quickcount</i> oleh <i>surveyor</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Surveyor</i> melakukan login dengan mengisi kolom <i>username</i> dengan kata surver, kolom <i>password</i> dengan kata surver, dan menekan tombol <i>login</i> 2. <i>Surveyor</i> masuk ke halaman input hasil dengan menekan tombol input hasil 3. <i>Surveyor</i> masuk ke halaman <i>new record</i> dengan menekan tombol <i>new record</i> 4. <i>Surveyor</i> memasukkan data pada form <i>input</i> yang disediakan yang berupa periode yang diisi dengan periode 2013-2018, TPS dikosongkan, Kandidat diisi dengan pasangan dwi-udin, jumlah suara 12 5. <i>Surveyor</i> menyimpan data hasil <i>quickcount</i> dengan menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem tidak dapat memasukkan data hasil <i>quickcount</i>
Hasil yang didapatkan	Sistem tidak dapat melakukan penambahan data hasil <i>quickcount</i>
Status Validitas	Valid

Pada Tabel 6.1 merupakan kasus uji validasi penambahan data hasil non ideal quickcount yang bertujuan untuk menambahkan data hasil *quickcount* oleh *surveyor*. Dalam hal ini prosedur uji yang dilakukan ketika *surveyor* melakukan login dengan mengisi kolom *username* dengan kata surver, kolom *password* dengan kata surver, dan menekan tombol *login*. Kemudian *Surveyor* masuk ke halaman input hasil dengan menekan tombol input hasil. *Surveyor* masuk ke halaman *new record* dengan menekan tombol *new record*. *Surveyor* memasukkan data pada form *input* yang disediakan yang berupa periode yang diisi dengan periode 2013-2018, kolom TPS dikosongkan, Kandidat diisi dengan pasangan

dwi-udin, jumlah suara 12. *Surveyor* menyimpan data hasil *quickcount* dengan menekan tombol *save*. Hasil yang diharapkan dari prosedur uji agar sistem tidak dapat memasukkan data hasil *quickcount*. Hasil yang didapatkan sistem tidak dapat menambahkan data hasil *quickcount*. Sehingga kasus uji validasi penambahan hasil *quickcount* dalam kondisi non ideal telah valid.

6.1.1.3. Kasus Uji Validasi Penghapusan Data Hasil *Quickcount*

Tabel 6. 2 Kasus Uji Validasi Penghapusan Data Hasil *Quickcount*

Nama Kasus Uji	Penghapusan Data Hasil <i>Quickcount</i>
Tujuan Pengujian	Untuk menghapus data hasil <i>quickcount</i> oleh <i>surveyor</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Surveyor</i> melakukan login dengan mengisi kolom <i>username</i> dengan kata server, kolom <i>password</i> dengan kata server, dan menekan tombol <i>login</i> 2. <i>Surveyor</i> masuk ke halaman data hasil <i>quickcount</i> dengan menekan tombol <i>input</i> hasil 3. <i>Surveyor</i> memberi tanda <i>checklist</i> pada data hasil <i>quickcount</i> pada TPS blimbing1 TPS1 nomor urut 1 yang akan dihapus 4. <i>Surveyor</i> menekan tombol <i>delete</i>
Hasil yang diharapkan	Sistem dapat menghapus data hasil <i>quickcount</i> dan menampilkan daftar data hasil <i>quickcount</i> yang masih tersimpan
Hasil yang didapatkan	Sistem menghapus data hasil <i>quickcount</i> pada TPS blimbing1 TPS1 nomor urut 1
Status Validitas	Valid

Pada Tabel 6.2 merupakan kasus uji validasi penghapusan data hasil *quickcount* yang bertujuan untuk menambahkan data hasil *quickcount* oleh *surveyor*. Dalam hal ini prosedur uji yang dilakukan ketika *surveyor* melakukan *login* dengan mengisi kolom *username* dengan kata server, kolom *password* dengan kata server, dan menekan tombol *login*. *Surveyor* masuk ke halaman data hasil *quickcount* dengan menekan tombol *input* hasil. *Surveyor* memberi tanda *checklist* pada data hasil *quickcount* pada TPS blimbing1 TPS1 nomor urut 1

yang akan dihapus. Surveyor menekan tombol *delete*. Hasil yang diharapkan dari prosedur uji agar sistem dapat menghapus data hasil *quickcount* dan menampilkan daftar data hasil *quickcount* yang masih tersimpan Hasil yang didapatkan sistem menghapus data hasil *quickcount* pada blimbing1 TPS1 pasangan nomor urut 1 dalam data hasil. Sehingga kasus uji validasi penghapusan hasil *quickcount* telah *valid*.

6.1.1.4. Kasus Uji Validasi Pengubahan Data Hasil *Quickcount*

Tabel 6. 3 Kasus Uji Validasi Pengubahan Data Hasil *Quickcount*

Nama Kasus Uji	Pengubahan Data Hasil <i>Quickcount</i>
Tujuan Pengujian	Untuk mengubah data hasil <i>quickcount</i> oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Surveyor</i> melakukan login dengan mengisi kolom <i>username</i> dengan kata server, kolom <i>password</i> dengan kata server, dan menekan tombol <i>login</i> 2. <i>Surveyor</i> masuk ke halaman data hasil <i>quickcount</i> dengan menekan tombol input hasil 3. <i>Surveyor</i> memberi tanda <i>checklist</i> pada data hasil <i>quickcount</i> blimbing 1 TPS 1 pada pasangan nomor urut 2 yang akan diubah 4. <i>Surveyor</i> menekan tombol <i>edit</i> pada kolom <i>action</i> 5. <i>Surveyor</i> mengubah data hasil <i>quickcount</i> baru kedalam form yang disediakan, berupa mengubah data perolehan jumlah suara menjadi 65 6. <i>Surveyor</i> menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem dapat mengubah data hasil <i>quickcount</i> dan menampilkan daftar data hasil <i>quickcount</i> yang telah diubah
Hasil yang didapatkan	Sistem mengubah data hasil <i>quickcount</i> blimbing 1 TPS 1 yang telah ditandai dan menampilkan data hasil <i>quickcount</i> perolehan jumlah suara menjadi 65

Status Validitas	Valid
------------------	-------

Pada Tabel 6.3 merupakan kasus uji validasi perubahan data hasil *quickcount* yang mempunyai tujuan untuk mengubah data hasil *quickcount* oleh *administrator*. Dengan melakukan prosedur uji ketika surveyor melakukan login dengan mengisi kolom username dengan kata server, kolom password dengan kata server, dan menekan tombol login. Surveyor masuk ke halaman data hasil *quickcount* dengan menekan tombol input hasil. Surveyor memberi tanda checklist pada data hasil *quickcount* blimbing 1 TPS 1 pada pasangan nomor urut 2 yang akan diubah. Surveyor menekan tombol edit pada kolom action. Surveyor mengubah data hasil *quickcount* baru kedalam form yang disediakan, berupa mengubah data perolehan jumlah suara menjadi 65. Surveyor menekan tombol save. Dengan mempunyai hasil yang diharapkan agar sistem dapat mengubah data hasil *quickcount* dan menampilkan daftar data hasil *quickcount* yang telah diubah. Hasil yang didapatkan sistem dapat mengubah data hasil *quickcount* blimbing 1 TPS 1 yang telah ditandai dan menampilkan data hasil *quickcount* perolehan jumlah suara menjadi 65. Sehingga kasus uji validasi mengubah hasil *quickcount* telah *valid*.

6.1.1.5. Kasus Uji Validasi Menampilkan Halaman Hasil Perolehan Suara *Quickcount*

Tabel 6. 4 Kasus Uji Validasi Menampilkan Halaman Hasil Perolehan Suara *Quickcount*

Nama Kasus Uji	Menampilkan Halaman Perolehan Suara <i>Quickcount</i>
Tujuan Pengujian	Untuk menampilkan hasil perolehan suara <i>quickcount</i> oleh <i>viewer</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Viewer</i> berada pada halaman utama 2. <i>Viewer</i> dapat melihat hasil perolehan suara <i>quickcount</i> 3. <i>Viewer</i> mensorot nama kandidat
Hasil yang diharapkan	<i>Viewer</i> dapat melihat hasil perolehan suara <i>quickcount</i> , visi misi kandidat

Hasil yang didapatkan	Sistem menampilkan hasil perolehan suara <i>quickcount</i> , dan visi misi kandidat
Status Validitas	Valid

Pada Tabel 6.2 merupakan kasus uji validasi menampilkan halaman hasil perolehan suara *quickcount*. Tujuan dari pengujian ini untuk menampilkan hasil perolehan suara *quickcount* yang mempunyai prosedur uji yang pertama viewer berada pada halaman utama. Viewer dapat melihat hasil perolehan suara *quickcount* dan viewer dapat mensorot nama kandidat. Hasil yang diharapkan viewer dapat melihat hasil perolehan suara *quickcount*, visi misi kandidat. Hasil yang didapatkan sistem dapat menampilkan hasil perolehan suara *quickcount*, dan visi misi kandidat. Sehingga kasus uji validasi menampilkan halaman hasil suara *quickcount* telah *valid*.

6.1.2 Pengujian Non Fungsional

Pengujian perangkat lunak yang kedua dengan menggunakan pengujian non fungsional. Pengujian non fungsional merupakan pengujian yang bertujuan untuk mengetahui tingkat kenyamanan pengguna dalam mengoperasikan sistem. Pada pengujian perangkat lunak ini menggunakan lembar kuisioner yang akan diisi oleh pengguna saat melakukan simulasi terhadap sistem perangkat lunak. Hasil dari kuisioner terdapat pada lampiran.

6.2 Analisis

Proses analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian sistem *quickcount*.

6.2.1 Analisis Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan membandingkan kesesuaian hasil kinerja sistem dengan daftar kebutuhan yang sudah ditentukan sebelumnya.

Hasil pengujian validasi dengan metode *Blackbox-Testing* adalah 100% dan dapat disimpulkan bahwa fungsionalitas kerja dalam sistem *quickcount* sudah

berjalan dengan baik dan sesuai dengan kebutuhan yang ditentukan dalam analisa kebutuhan.

6.2.2 Analisis Pengujian Non Fungsional

Proses analisis terhadap hasil pengujian non fungsional dilakukan dengan menyimpulkan jawaban dari lembar kuisioner yang telah dibagikan pada pengguna pada saat simulasi.

Hasil pengujian non fungsional dengan menggunakan kuisioner adalah 100% dan dapat disimpulkan bahwa kenyamanan pengguna pada saat menggunakan sistem perangkat lunak *quickcount* ini sudah memenuhi kebutuhan pengguna.



BAB VII PENUTUP

7.1. KESIMPULAN

Pengolahan data quickcount sebelumnya dilakukan dengan memasukkan data hasil quickcount secara manual dengan dibantu banyak surveyor dan hanya satu admin yang dapat mengolah data hasil. Hal ini membuat proses quickcount membutuhkan waktu seminggu dalam melakukan persiapan dan pengerjaannya. Sehingga penulis membuat proses quickcount dengan menggunakan sistem berbasis web. Sistem quickcount yang dibangun sesuai dengan proses bisnis yang dibutuhkan berdasarkan pengujian dengan menggunakan pengujian fungsional yang berupa uji validitas fungsional sistem.

Berdasarkan hasil observasi dengan menggunakan kuisisioner terhadap pengguna sistem, sebagian besar pengguna menyatakan bahwa sistem quickcount yang dibangun dapat lebih cepat dan efisien terhadap pengolahan data sesuai dengan kebutuhan pengguna. Hal ini berarti sistem quickcount dapat melakukan proses quickcount menjadi lebih mudah daripada proses sebelumnya.

7.2. SARAN

Untuk meningkatkan hasil yang telah dicapai dalam penelitian ini dapat dilakukan pengembangan sistem. Pengembangan sistem *quickcount* ini berfokus pada rancangan dan implementasi sistem untuk mengolah hasil *quickcount*, maka untuk pengembangan selanjutnya dapat ditambahkan unsur keamanan *confidentiality* dengan menggunakan metode enkripsi dengan algoritma kriptografi kunci-publik yaitu *Rivest Shamir Adleman* (RSA) agar data hasil *quickcount* mempunyai jaminan kerahasiaan dan terhindar dari pemalsuan data pada saat proses memasukkan hasil *quickcount*.



DAFTAR PUSTAKA

- [ST-05] Sutabri, Tata 2005. *Aplikasi PHP*. Yogyakarta : Andi
- [LSI-JIP] Lingkaran Survei Indonesia (LSI) & Jaringan Isu Publik (JIP). 2007.
< <http://www.lsi.co.id/>> diakses pada tanggal 10 September 2013 pkl. 11.44pm
- [SBim-03] Sunarfrihanto, Bimo, 2003. *PHP dan MySQL untuk Web*, Yogyakarta :Andi
- [KSM-07] Kismiantini. 2007. *Pengumpulan Data dengan Quick Count dan Exit Poll*. Yogyakarta
- [GJM-04] J.Myers, Glendford., 2004, "The Art of Software Testing, Second Edition", John Wiley & Sons,Inc.,Hoboken,New Jersey
- [SB-12] Sidik,Betha. 2012."Framework Code Igniter", Penerbit : Informatika Bandung
- [PM-03] Priestley, Mark. 2003. "Practical Object- Oriented Design With UML", Edisi 2, Mcgraw Hill Education
- [ERI-06] Eriyanto, 2006. "Panduan Penyelenggaraan *Quickcount*". Penerbit : Lingkaran Survey Indonesia, Jakarta
- [SMJ-08] Mujani, Saiful. 2008. "Memahami Hakekat *Quickcount*". Penerbit : Sumatera Exspress
- [ESTOK-2002] Estok M, Nevite N & Cowan G, 2002. *The Quickcount and Election Observation*, Washington : NDI.

LAMPIRAN

Lampiran 1. Deskripsi Perancangan Use Case login

Skenario Kasus pada Sistem	
Nama	<i>Login</i>
Tujuan	<i>Administrator</i> atau <i>surveyor</i> dapat melakukan <i>login</i>
Deskripsi	<i>Usecase</i> ini menjelaskan bagaimana <i>administrator</i> atau <i>surveyor</i> dapat melakukan <i>login</i>
Aktor	<i>Administrator</i> dan <i>Suveyor</i>
Skenario Utama	
Kondisi awal	<i>Administrator</i> atau <i>surveyor</i> telah berada pada halaman utama
Aksi Aktor	Reaksi Sistem
<i>Administrator</i> atau <i>surveyor</i> memasukkan username dan password pada form login, dan kemudian menekan tombol login	Sistem akan menampilkan halaman utama <i>administrator</i> atau <i>surveyor</i>
Kondisi Akhir	Sistem berada pada halaman utama bagi <i>administrator</i> dan <i>surveyor</i>

Lampiran 2 Deskripsi Perancangan Use Case Olah User

Skenario Kasus pada Sistem	
Nama	Olah <i>User</i>
Tujuan	<i>Administrator</i> dapat melakukan pendaftaran <i>username</i> dan <i>password</i> bagi <i>surveyor</i>
Deskripsi	<i>Usecase</i> ini menjelaskan bagaimana <i>administrator</i> melakukan pendaftaran <i>username</i> dan <i>password</i> bagi <i>surveyor</i>
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi awal	<i>Administrator</i> telah <i>login</i> dan menekan tautan olah data <i>user</i> pada halaman utama <i>Administrator</i>
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> - <i>Administrator</i> menekan tautan olah data <i>user</i> dihalaman <i>Administrator</i> - <i>Administrator</i> memasukkan data pendaftaran <i>user</i> dengan menekan tombol tambah <i>user</i> - Jika <i>Administrator</i> ingin menghapus salah satu <i>user</i> dengan menekan tombol <i>delete</i> 	<ul style="list-style-type: none"> - Sistem akan menampilkan halaman olah data <i>user</i> - Sistem akan melakukan input data <i>user</i> dan menyimpannya - Sistem akan menghapus data <i>user</i> yang telah dipilih untuk dihapus
Kondisi Akhir	Sistem akan menyimpan data dan menampilkan data <i>user</i> yang sudah terdaftar

Lampiran 3 Deskripsi Perancangan Use Case Olah Kandidat

Skenario Kasus pada Sistem	
Nama	Olah kandidat
Tujuan	<i>Administrator</i> dapat melakukan olah data kandidat
Deskripsi	<i>Usecase</i> ini menjelaskan bagaimana <i>administrator</i> dapat menambah, mengubah, dan menghapus daftar kandidat
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi awal	<i>Administrator</i> telah <i>login</i> dan menekan tautan olah data kandidat pada halaman utama <i>Administrator</i>
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> - <i>Administrator</i> menekan tautan olah data kandidat dihalaman <i>Administrator</i> - <i>Administrator</i> memasukkan data kandidat dengan menekan tombol simpan - <i>Administrator</i> dapat mengubah informasi data kandidat dengan memilih salah satu kandidat dan menekan tombol <i>edit</i> - Jika <i>Administrator</i> ingin menghapus salah satu kandidat dengan menekan tombol hapus 	<ul style="list-style-type: none"> - Sistem akan menampilkan halaman olah data kandidat - Sistem akan melakukan input data kandidat dan menyimpannya - Sistem akan menampilkan data kandidat yang ingin diubah dan kemudian meng-<i>update</i> data kandidat yang telah diubah - Sistem akan menghapus data kandidat yang telah dipilih untuk dihapus
Kondisi Akhir	Sistem akan menyimpan data dan menampilkan data kandidat yang sudah ditambah/dirubah ataupun dihapus

Lampiran 4 Deskripsi Perancangan Use Case Kecamatan

Skenario Kasus pada Sistem	
Nama	Olah kecamatan
Tujuan	<i>Administrator</i> dapat melakukan olah data kecamatan
Deskripsi	<i>Usecase</i> ini menjelaskan bagaimana <i>administrator</i> dapat menambah, mengubah, dan menghapus daftar kecamatan
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi awal	<i>Administrator</i> telah <i>login</i> dan menekan tautan olah data kecamatan pada halaman utama <i>Administrator</i>
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> - <i>Administrator</i> menekan tautan olah data kecamatan dihalaman <i>Administrator</i> - <i>Administrator</i> memasukkan data kecamatan dengan menekan tombol simpan - <i>Administrator</i> dapat mengubah informasi data kecamatan dengan memilih salah satu kandidat dan menekan tombol <i>edit</i> - Jika <i>Administrator</i> ingin menghapus salah satu kecamatan dengan menekan tombol hapus 	<ul style="list-style-type: none"> - Sistem akan menampilkan halaman olah data kecamatan - Sistem akan melakukan input data kecamatan dan menyimpannya - Sistem akan menampilkan data kecamatan yang ingin diubah dan kemudian meng-<i>update</i> data kecamatan yang telah diubah - Sistem akan menghapus data kecamatan yang telah dipilih untuk dihapus
Kondisi Akhir	Sistem akan menyimpan data dan menampilkan data kecamatan yang sudah ditambah/dirubah ataupun dihapus



Lampiran 5 Deskripsi Perancangan Use Case Periode

Skenario Kasus pada Sistem	
Nama	Olah Periode
Tujuan	<i>Administrator</i> dapat melakukan olah data Periode
Deskripsi	<i>Usecase</i> ini menjelaskan bagaimana <i>administrator</i> dapat menambah, mengubah, dan menghapus daftar Periode
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi awal	<i>Administrator</i> telah <i>login</i> dan menekan tautan olah data Periode pada halaman utama <i>Administrator</i>
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> - <i>Administrator</i> menekan tautan olah data Periode dihalaman <i>Administrator</i> - <i>Administrator</i> memasukkan data Periode dengan menekan tombol simpan - <i>Administrator</i> dapat mengubah informasi data Periode dengan memilih salah satu Periode dan menekan tombol <i>edit</i> - Jika <i>Administrator</i> ingin menghapus salah satu Periode dengan menekan tombol hapus 	<ul style="list-style-type: none"> - Sistem akan menampilkan halaman olah data Periode - Sistem akan melakukan input data Periode dan menyimpannya - Sistem akan menampilkan data Periode yang ingin diubah dan kemudian meng-<i>update</i> data Periode yang telah diubah - Sistem akan menghapus data Periode yang telah dipilih untuk dihapus



Kondisi Akhir	Sistem akan menyimpan data dan menampilkan data Periode yang sudah ditambah/dirubah ataupun dihapus
---------------	---

Lampiran 6 Deskripsi Perancangan Use Case TPS

Skenario Kasus pada Sistem	
Nama	Olah TPS
Tujuan	<i>Administrator</i> dapat melakukan olah data TPS
Deskripsi	<i>Usecase</i> ini menjelaskan bagaimana <i>administrator</i> dapat menambah, mengubah, dan menghapus daftar TPS
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi awal	<i>Administrator</i> telah <i>login</i> dan menekan tautan olah data TPS pada halaman utama <i>Administrator</i>
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> - <i>Administrator</i> menekan tautan olah data TPS dihalaman <i>Administrator</i> - <i>Administrator</i> memasukkan data TPS dengan menekan tombol simpan - <i>Administrator</i> dapat mengubah informasi data TPS dengan memilih salah satu TPS dan menekan tombol <i>edit</i> - Jika <i>Administrator</i> ingin menghapus salah satu TPS dengan menekan tombol hapus 	<ul style="list-style-type: none"> - Sistem akan menampilkan halaman olah data TPS - Sistem akan melakukan input data TPS dan menyimpannya - Sistem akan menampilkan data TPS yang ingin diubah dan kemudian meng-<i>update</i> data TPS yang telah diubah - Sistem akan menghapus data TPS yang telah dipilih untuk dihapus

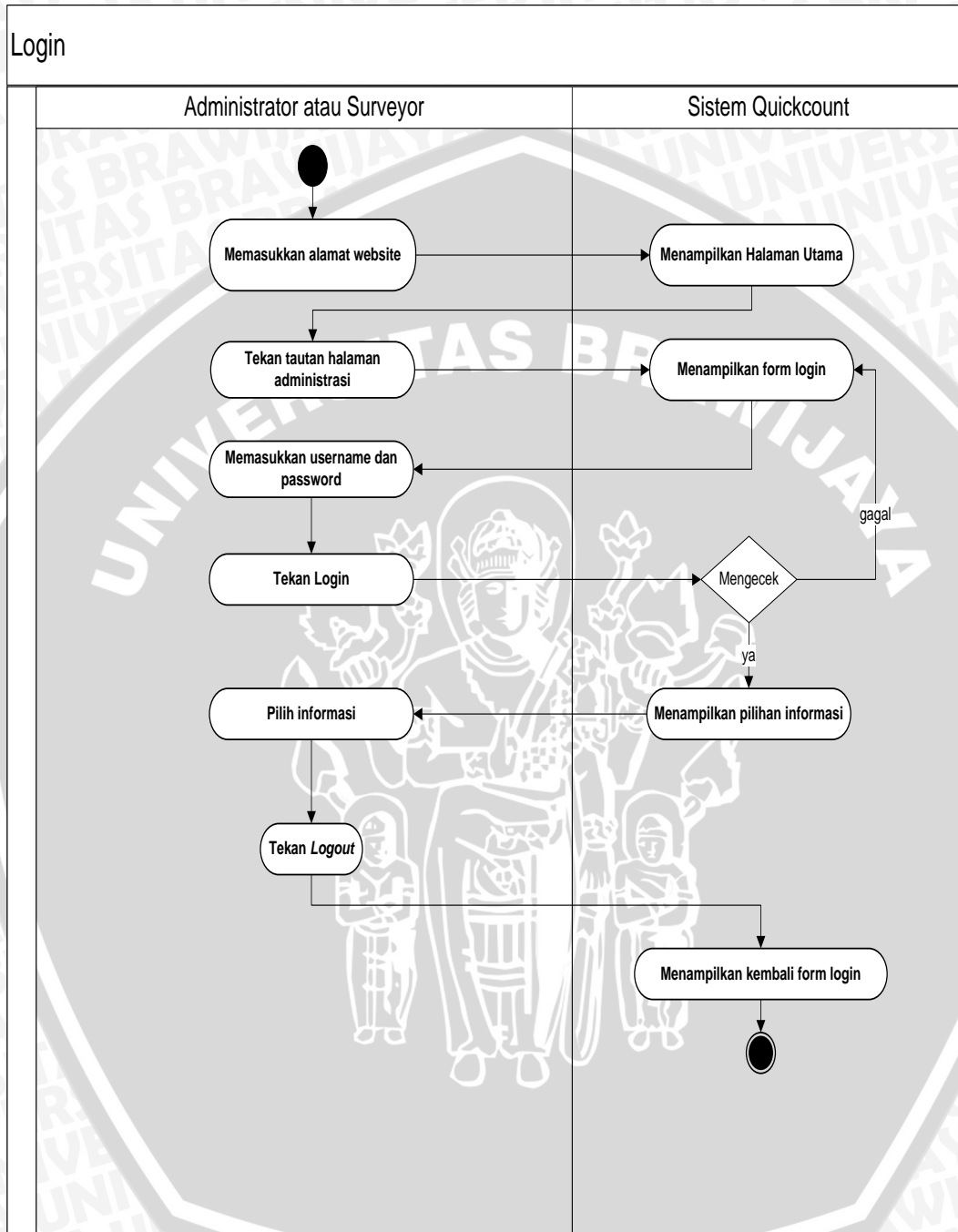
Kondisi Akhir	Sistem akan menyimpan data dan menampilkan data TPS yang sudah ditambah/dirubah ataupun dihapus
---------------	---

Lampiran 7 Deskripsi Perancangan Use Case Lihat Log Quickcount

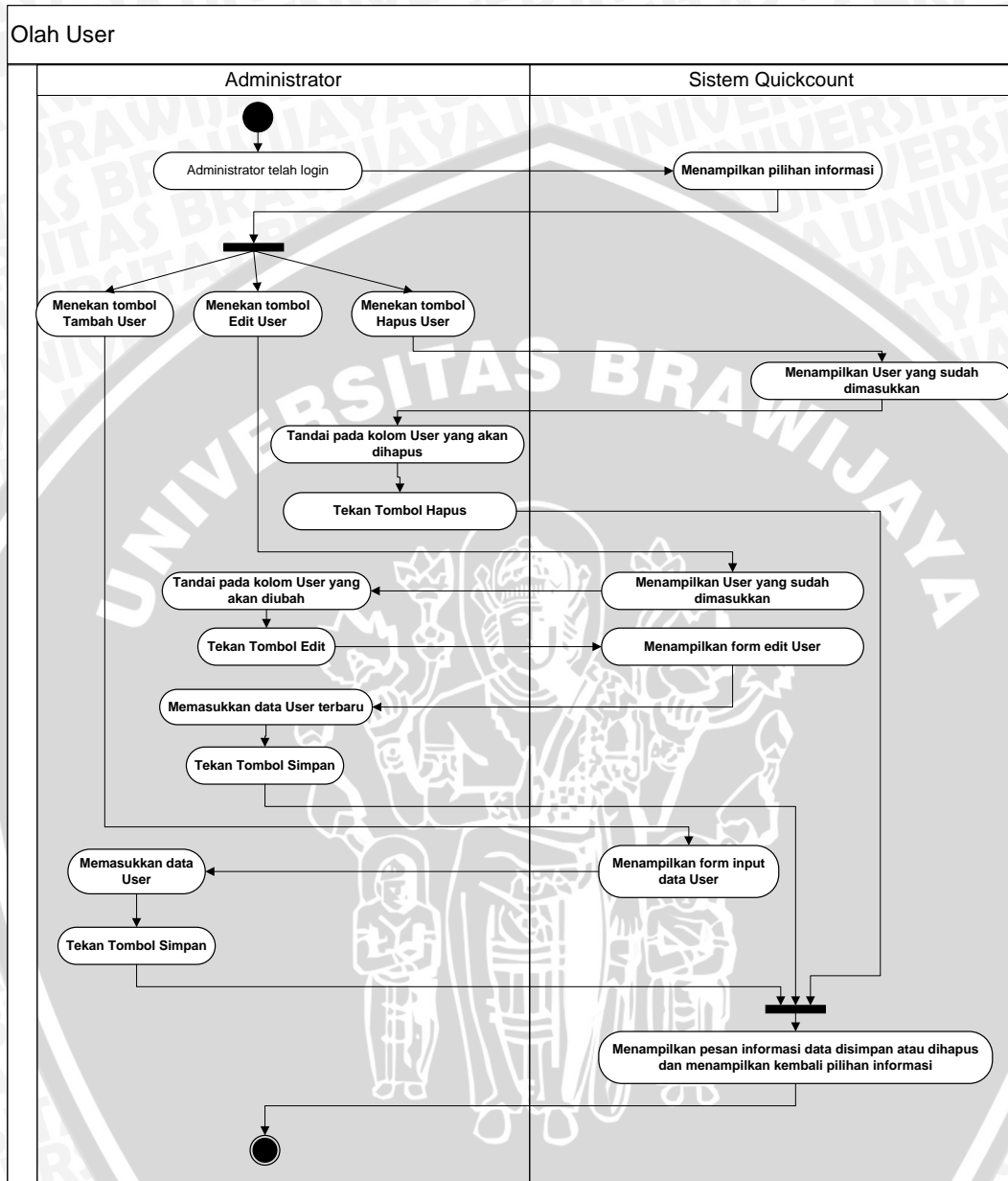
Skenario Kasus pada Sistem	
Nama	Lihat <i>log quickcount</i>
Tujuan	<i>Administrator</i> dapat melihat data hasil <i>quickcount</i> yang telah dimasukkan oleh <i>surveyor</i>
Deskripsi	<i>Usecase</i> ini menjelaskan bagaimana <i>administrator</i> dapat melihat data hasil <i>quickcount</i> yang telah dimasukkan oleh masing-masing <i>surveyor</i>
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi awal	<i>Administrator</i> telah <i>login</i> dan berada pada halaman utama <i>administrator</i>
Aksi Aktor	Reaksi Sistem
- <i>Administrator</i> menekan tautan lihat <i>log quickcount</i> dihalaman <i>Administrator</i>	- Sistem akan menampilkan halaman <i>log quickcount</i>
Kondisi Akhir	Sistem akan menampilkan halaman <i>log quickcount</i>



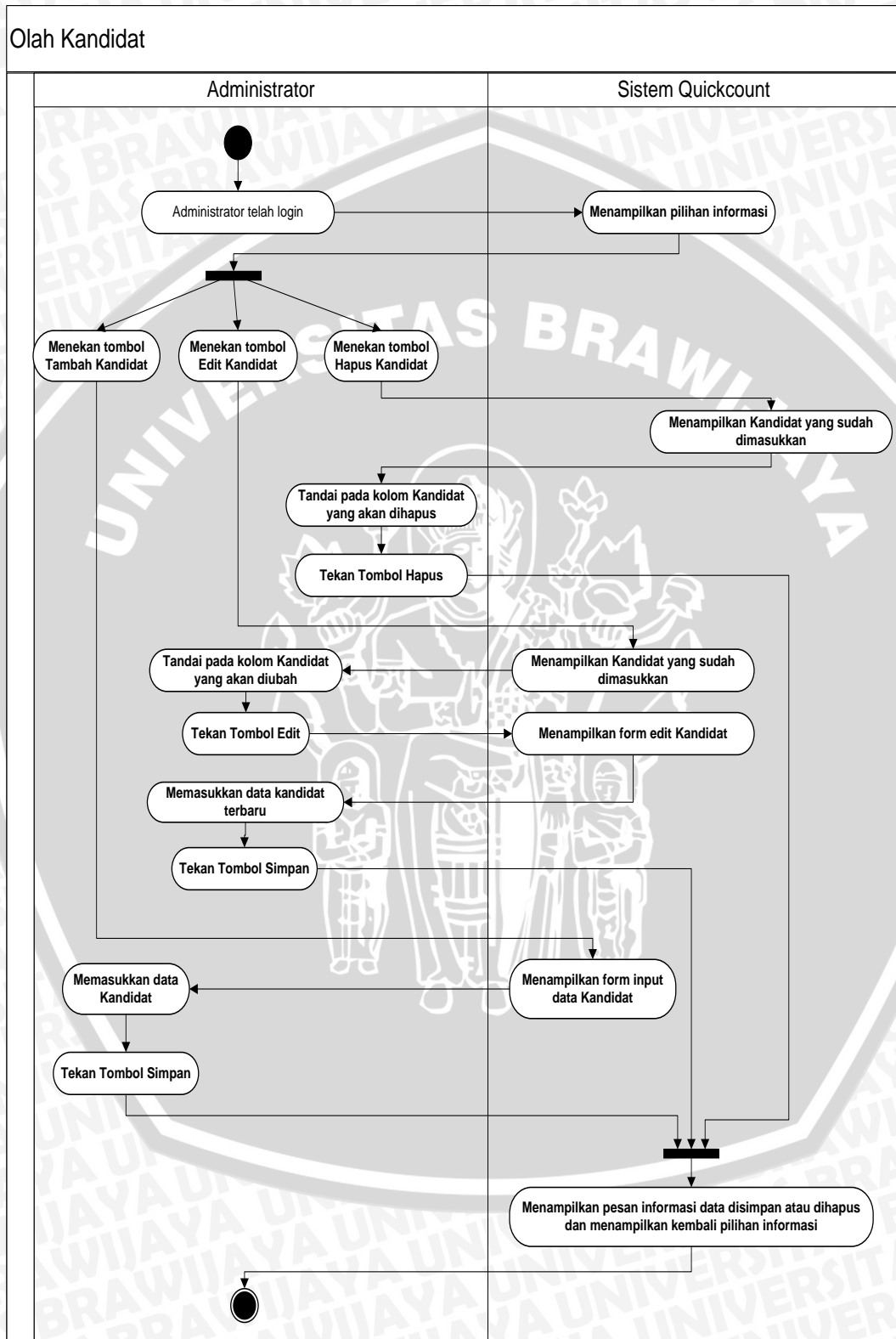
Lampiran 8 Activity Diagram Login



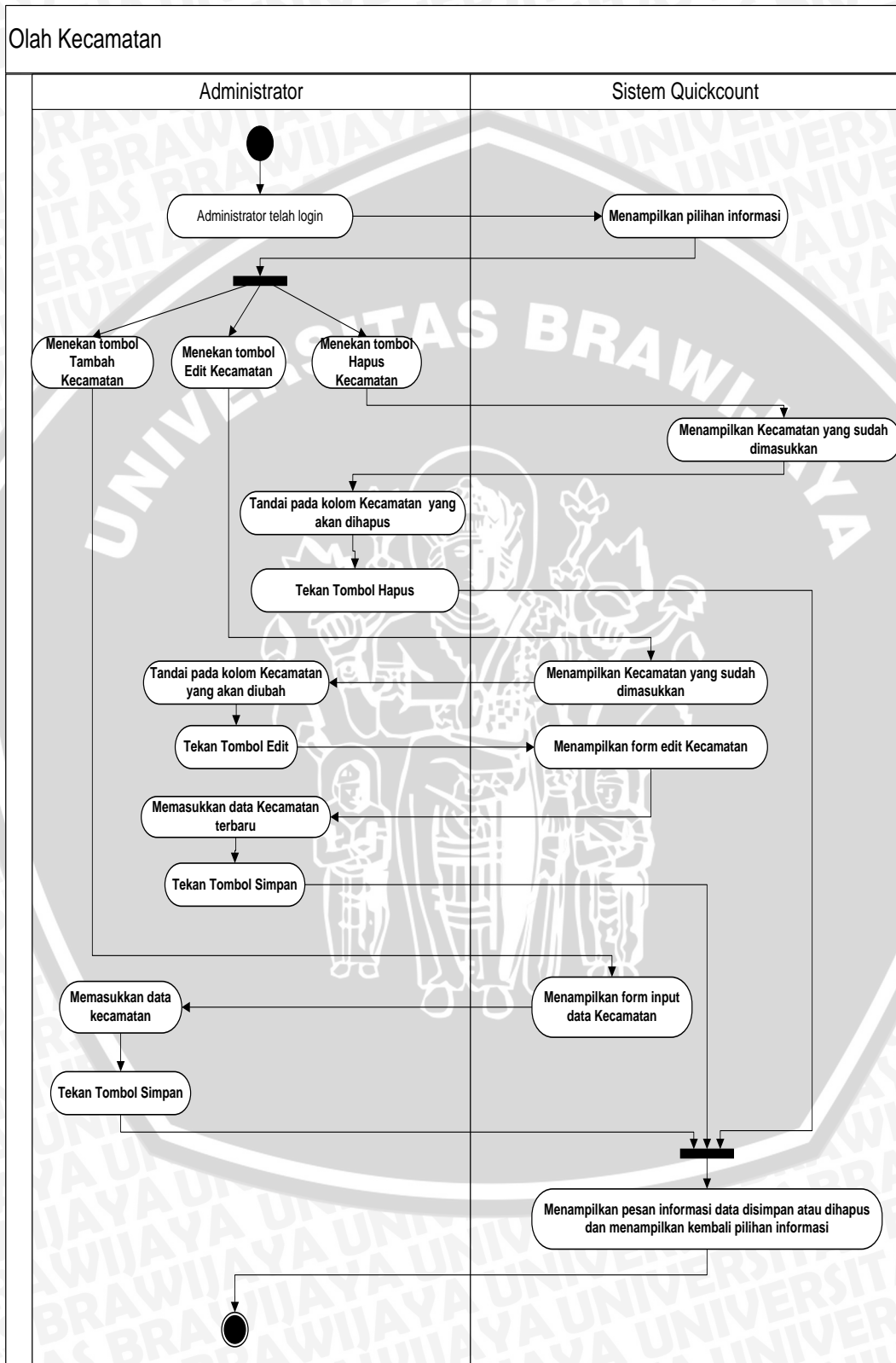
Lampiran 9 Activity Diagram Olah User



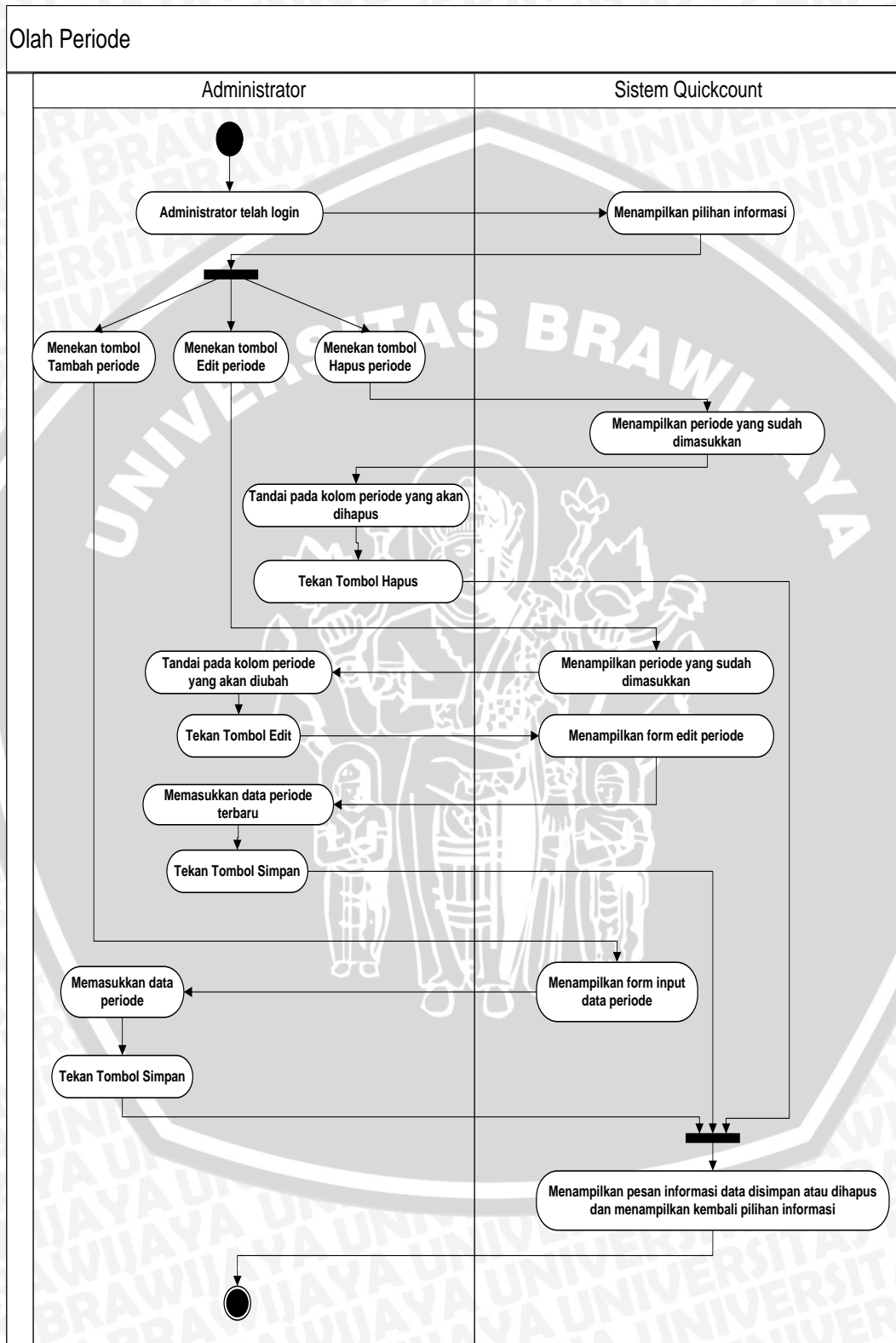
Lampiran 10 Activity Diagram Olah Kandidat



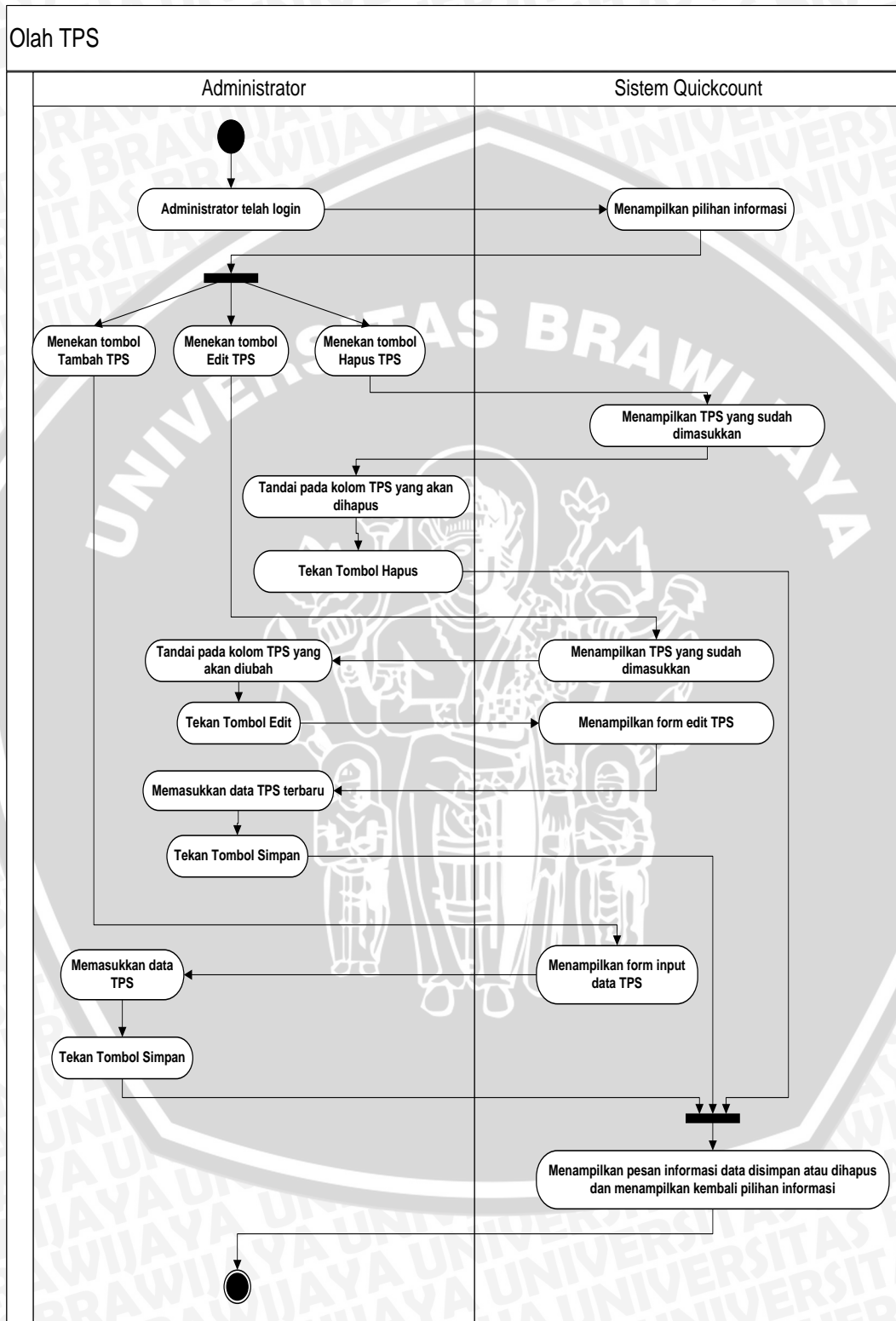
Lampiran 11 Activity Diagram Olah Kecamatan



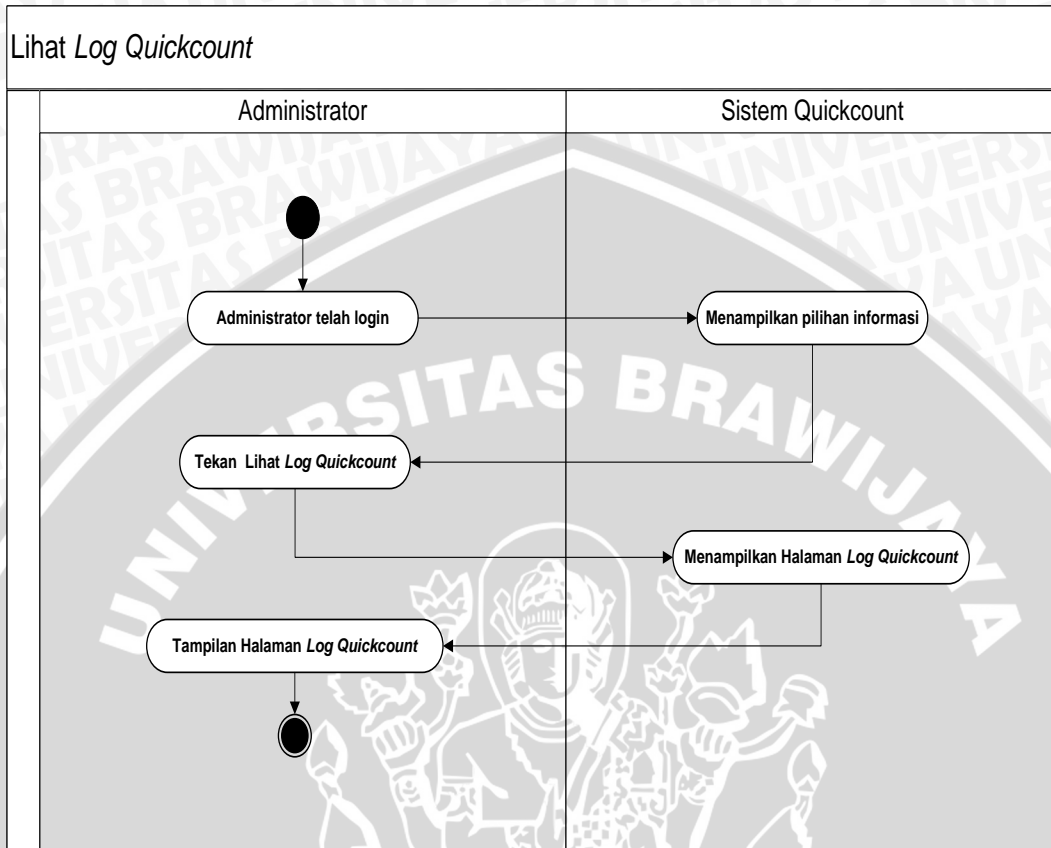
Lampiran 12 Activity Diagram Olah Periode



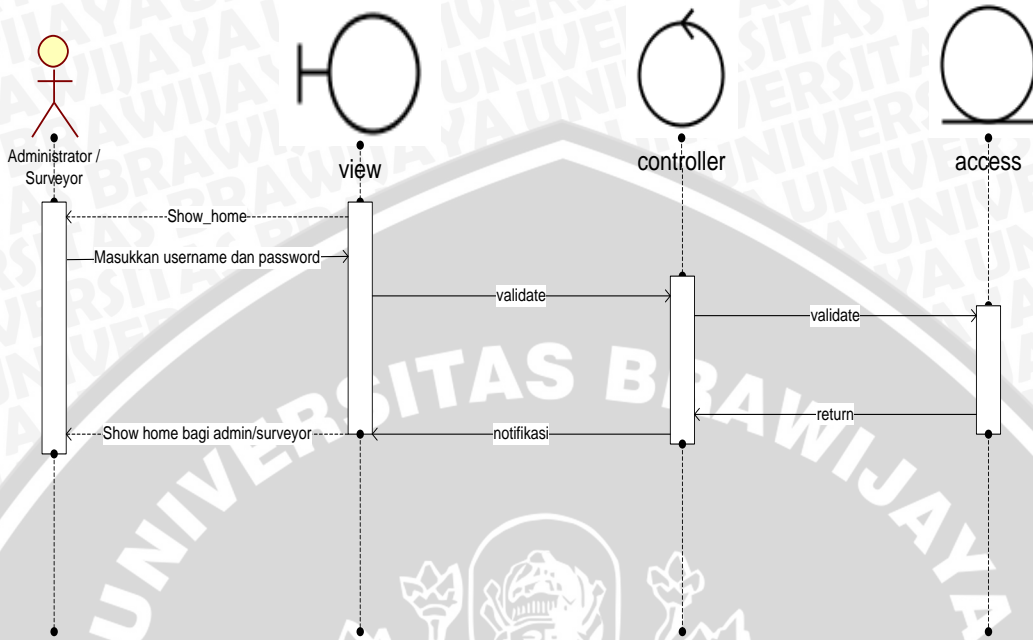
Lampiran 13 Activity Diagram Olah TPS



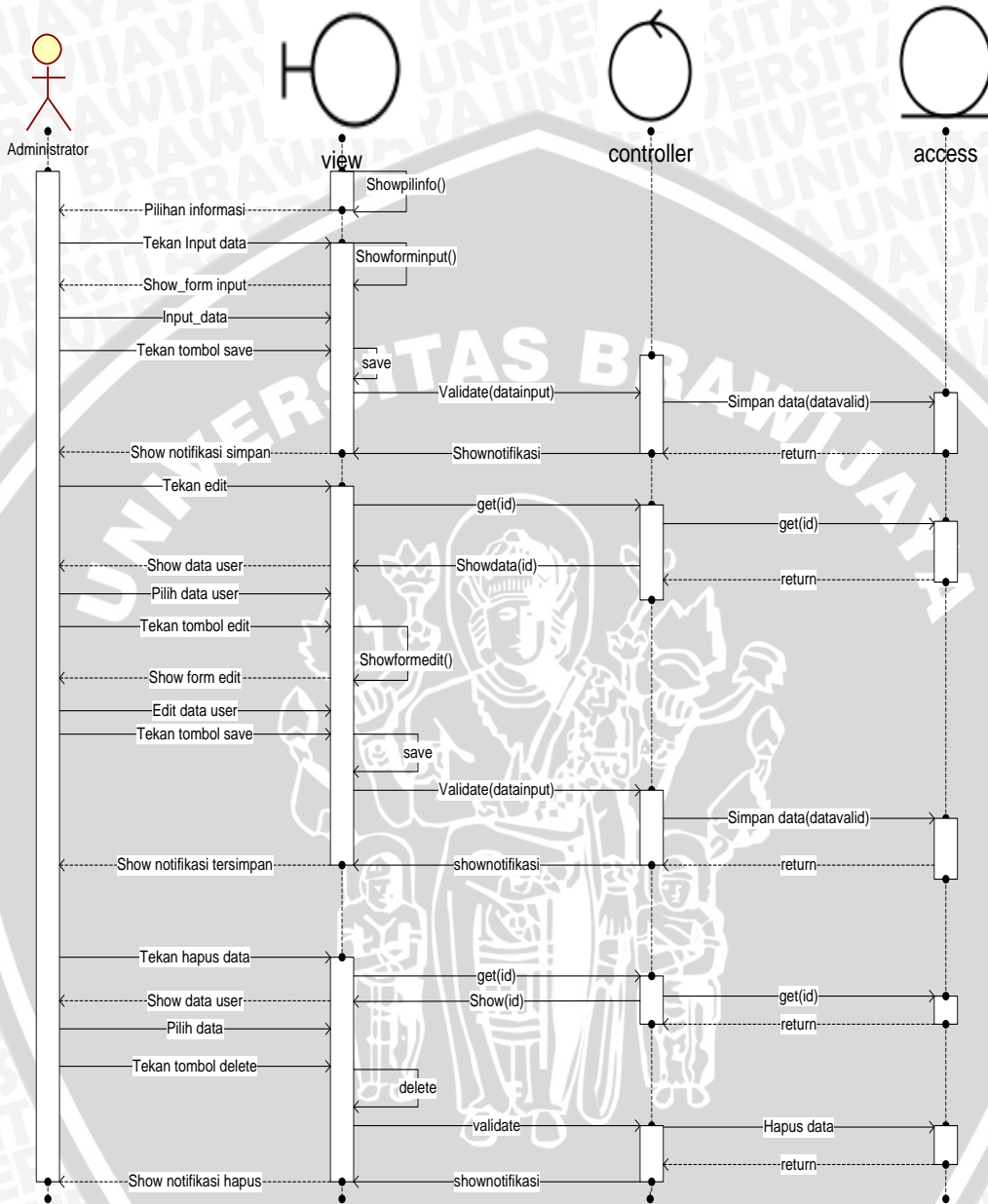
Lampiran 14 Activity Diagram Log Quickcount



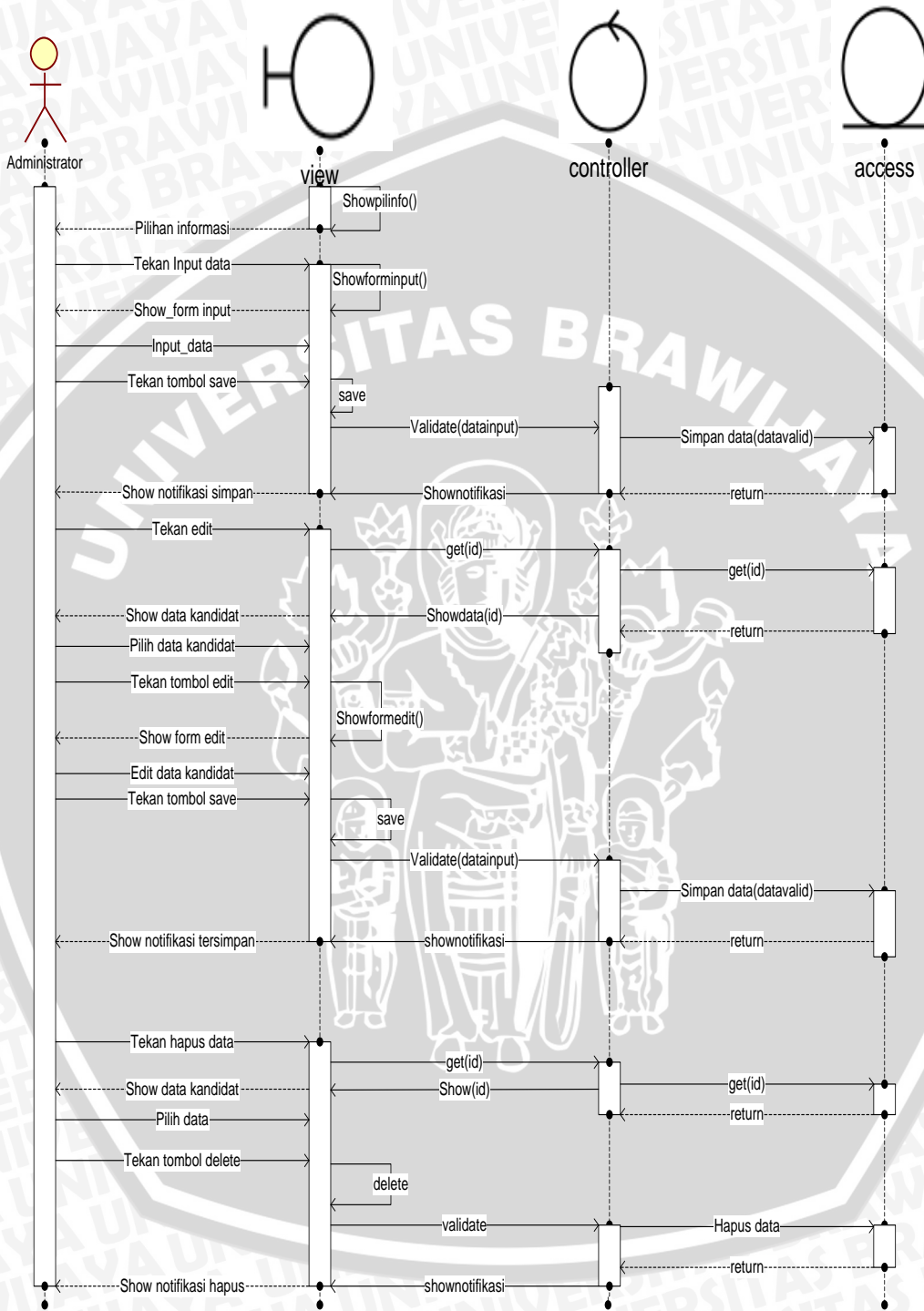
Lampiran 15 Sequence diagram Login



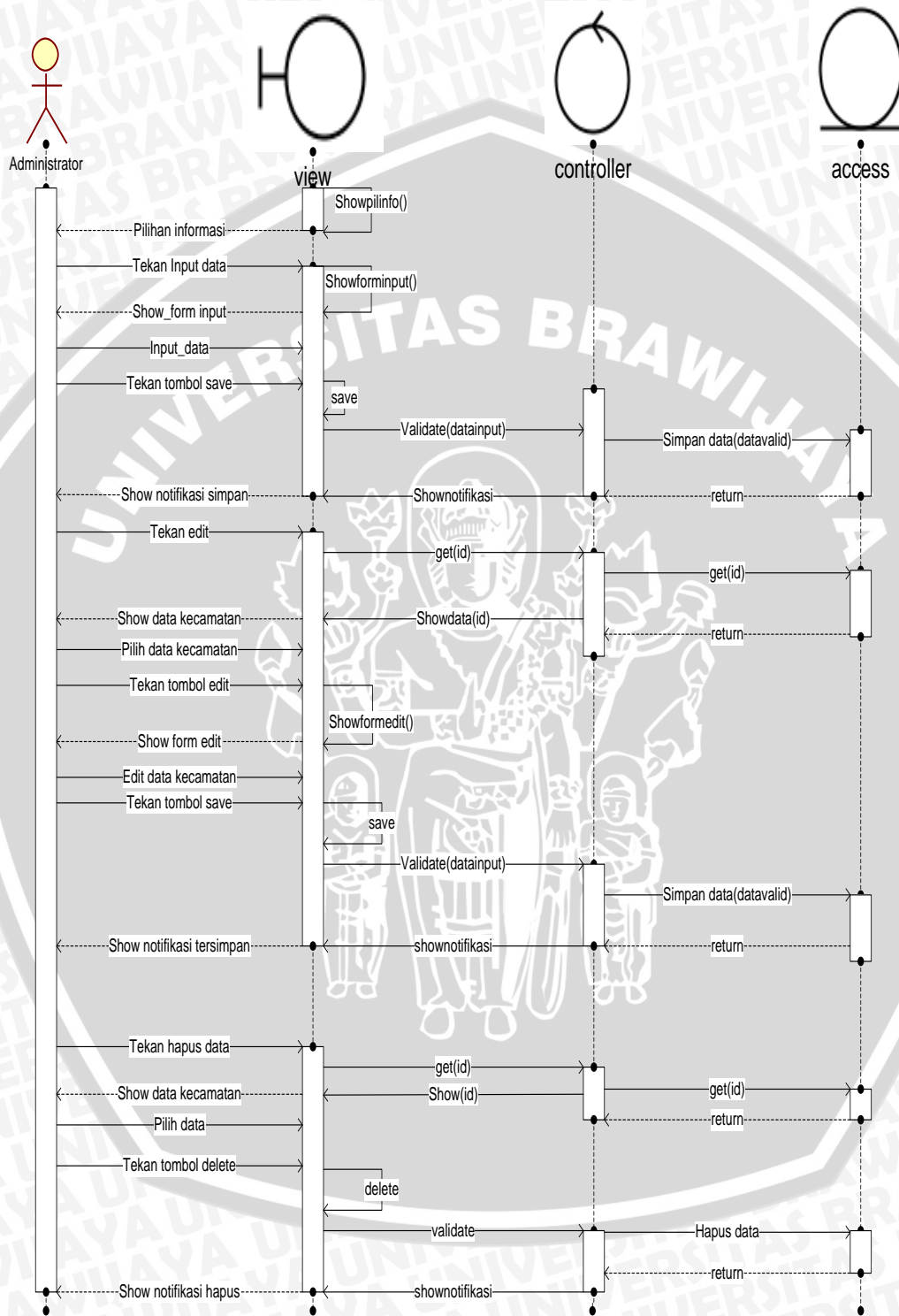
Lampiran 16 Sequence diagram Olah User



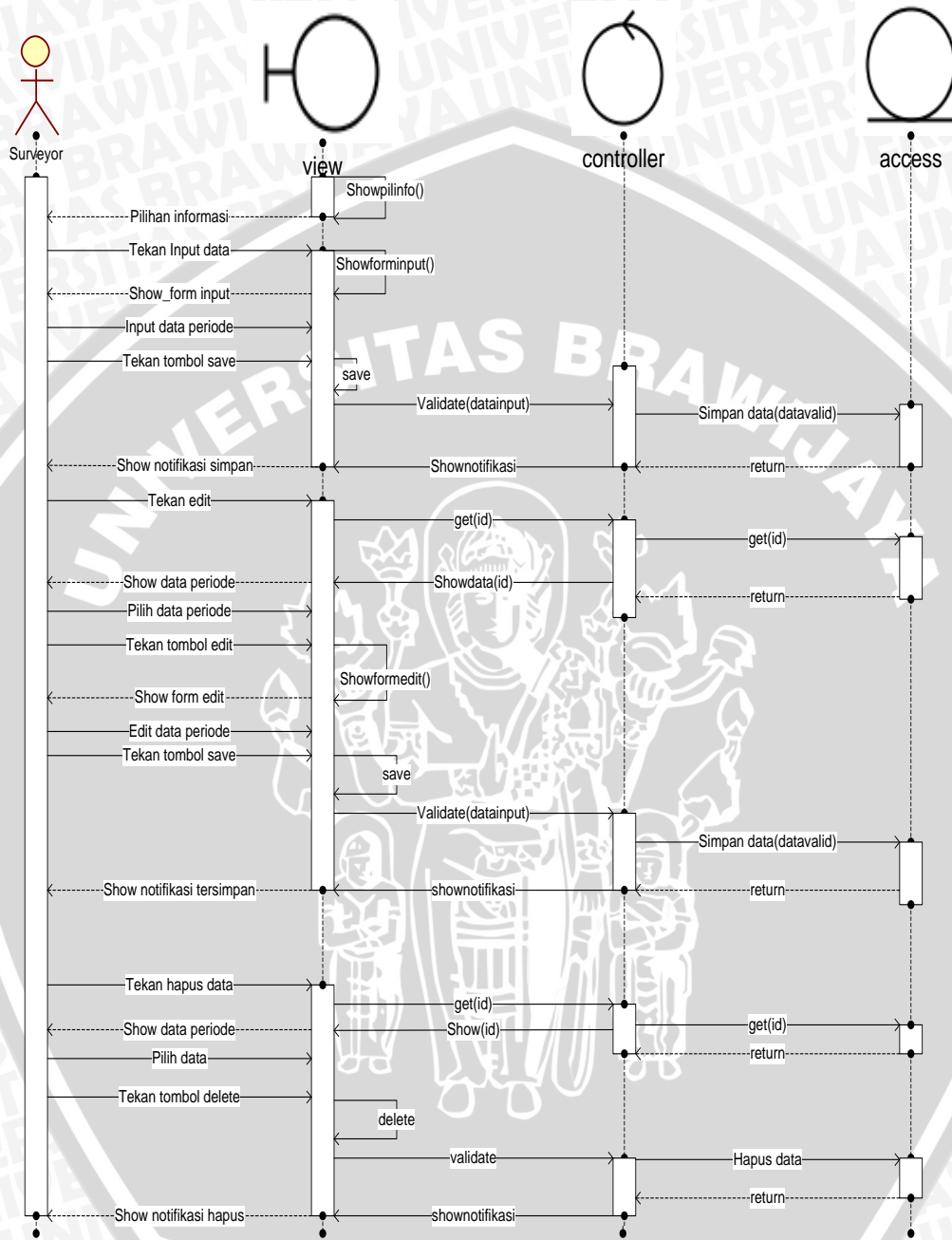
Lampiran 17 Sequence diagram Olah Kandidat



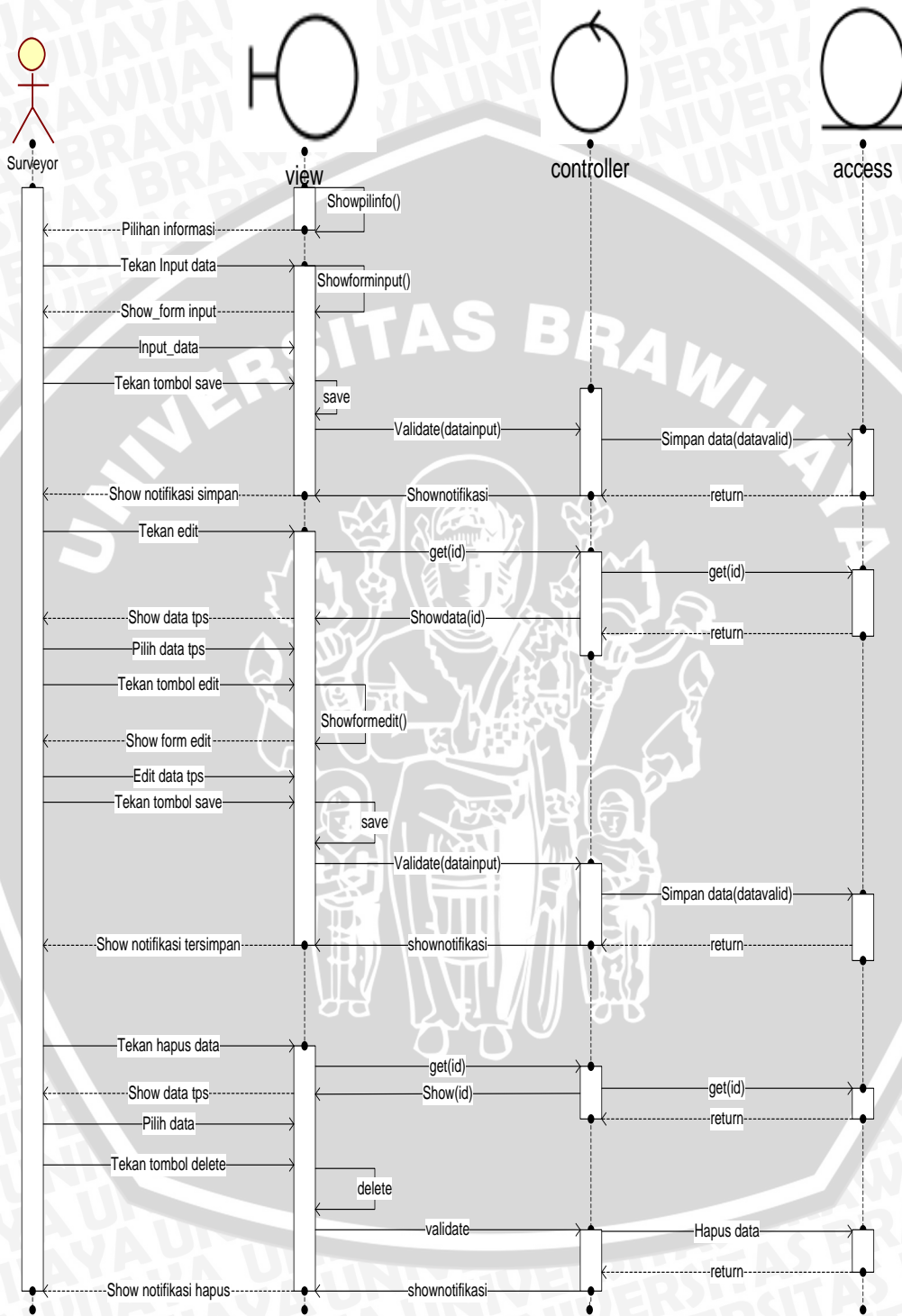
Lampiran 18 Sequence diagram Kecamatan



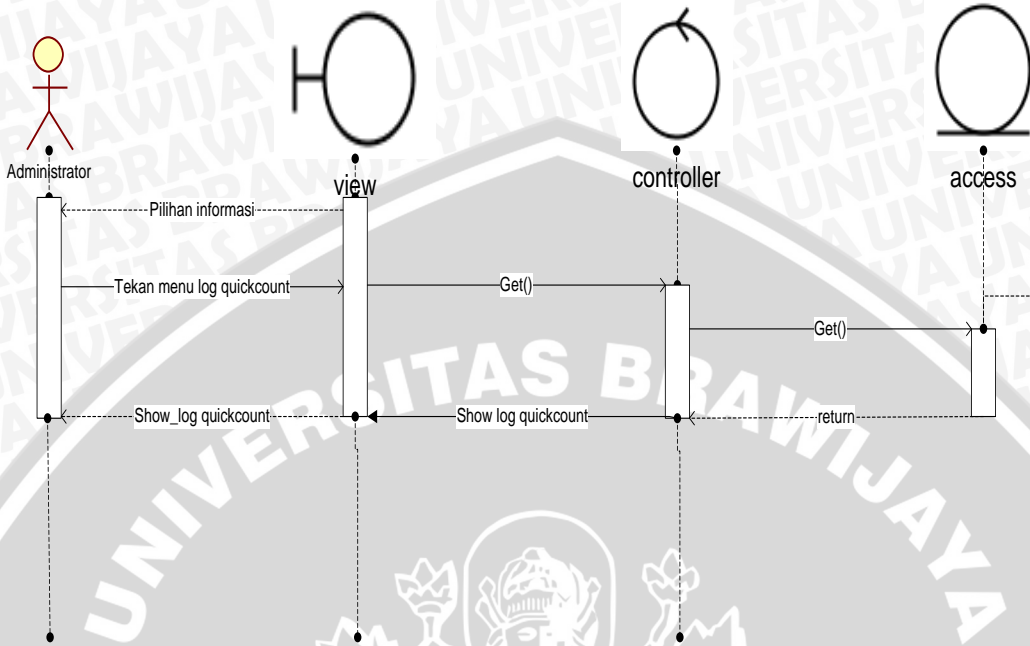
Lampiran 19 Sequence diagram Periode



Lampiran 20 Sequence diagram TPS



Lampiran 21 Sequence diagram Log Quickcount



Lampiran 22 Perancangan Antarmuka Login

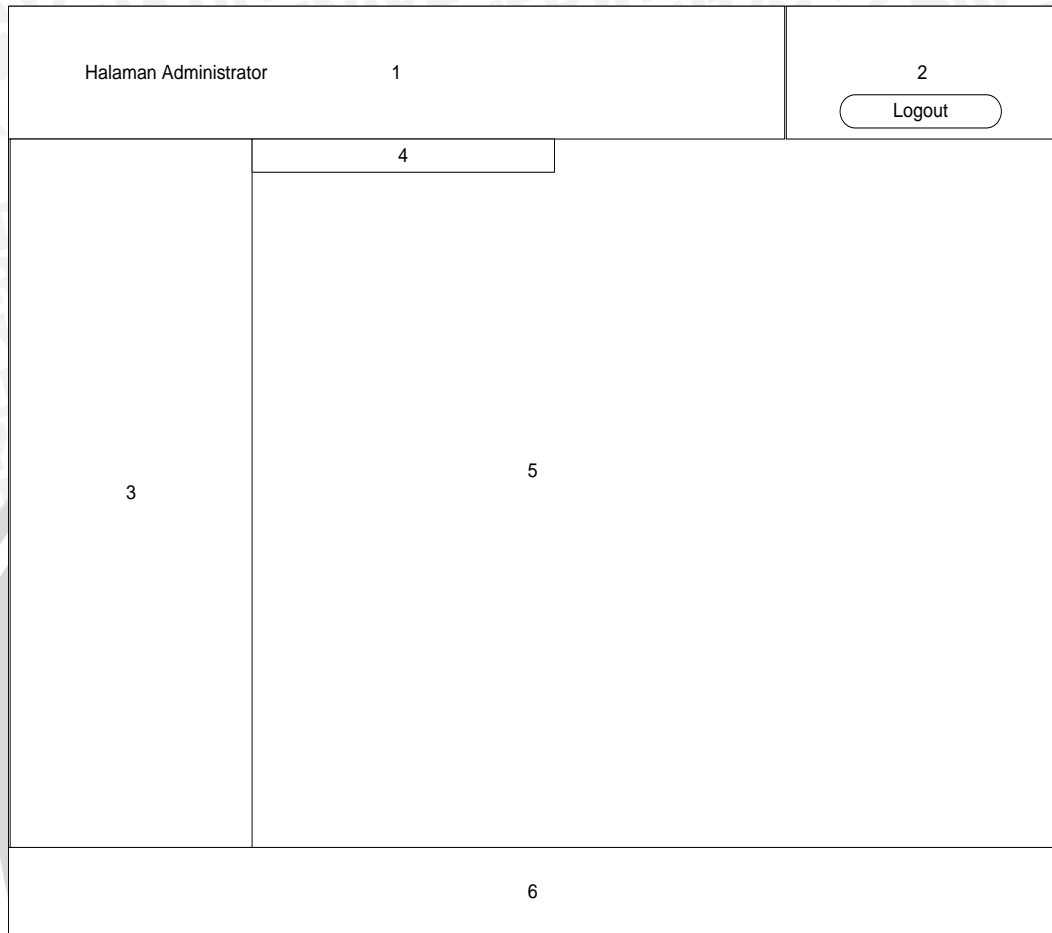
1

2

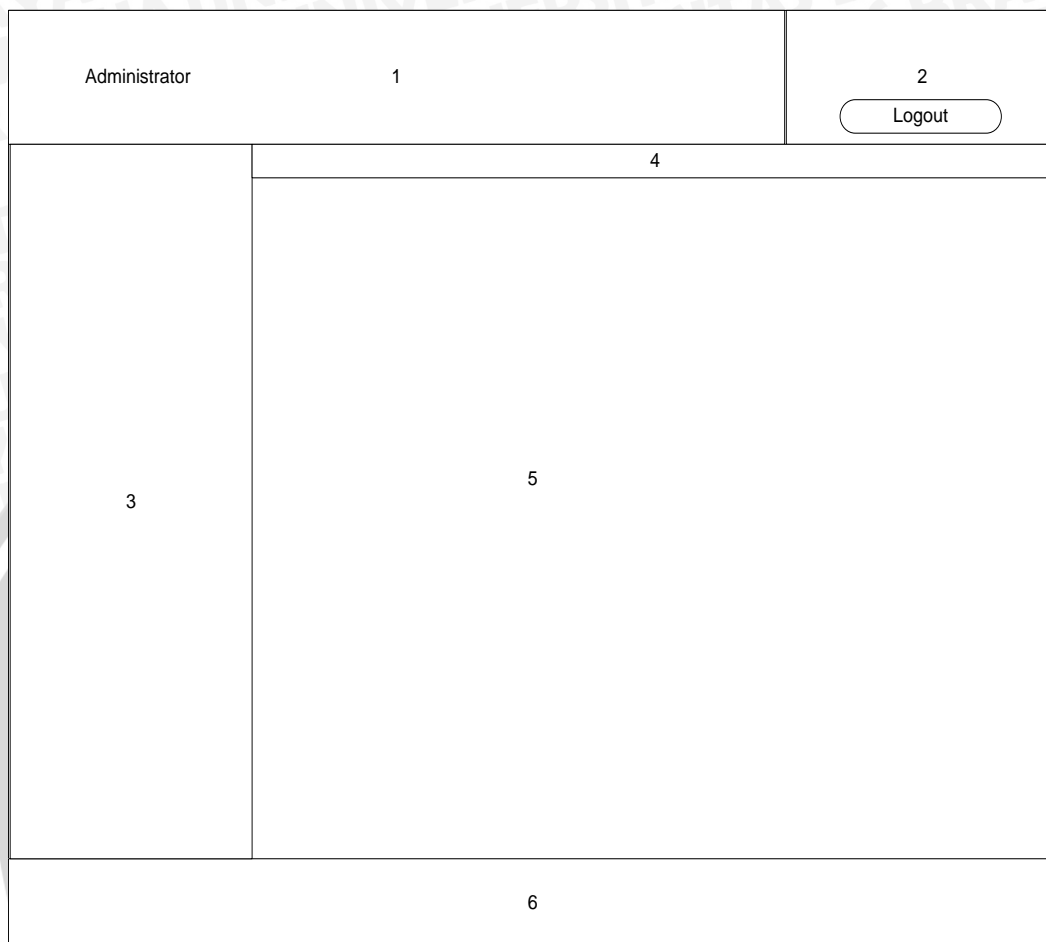
username	<input type="text"/>	
password	<input type="password"/>	
<input type="button" value="login"/>	<input type="button" value="reset"/>	<input type="button" value="home"/>



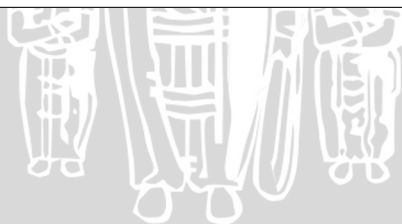
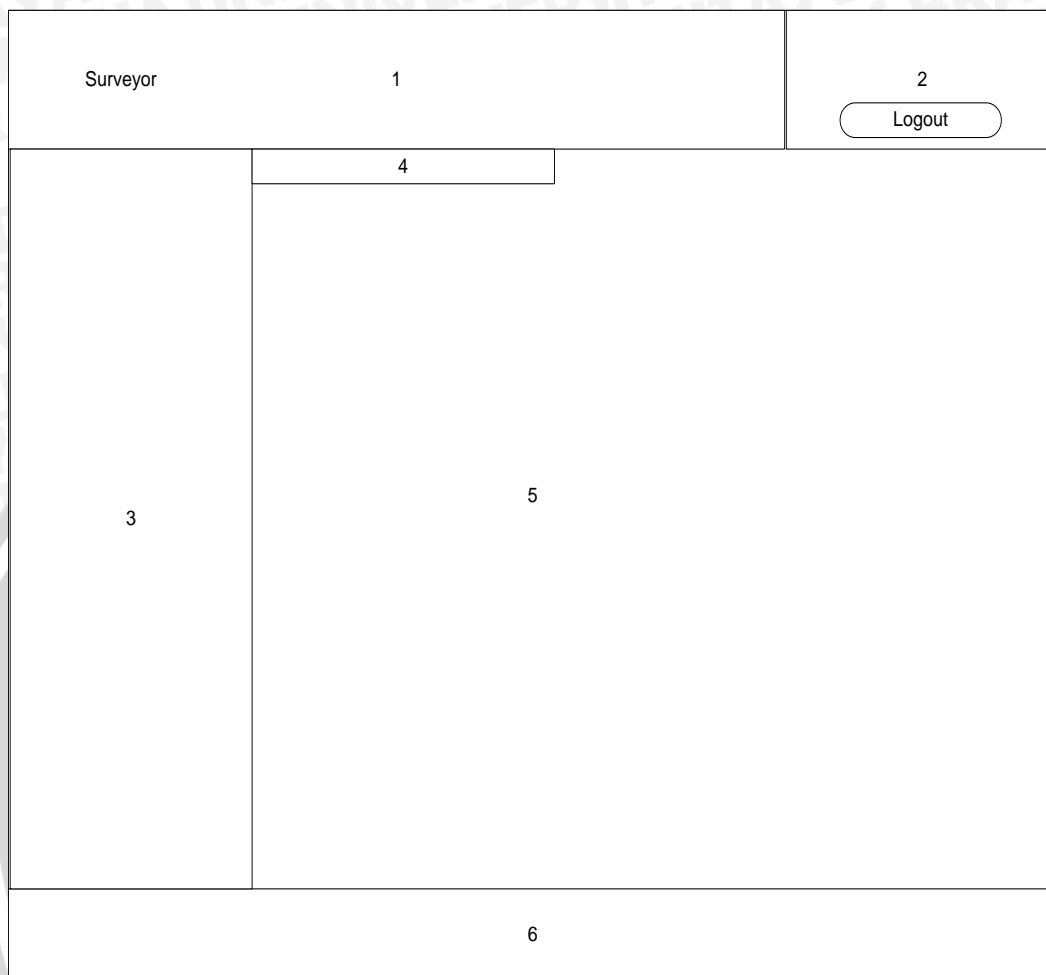
Lampiran 23 Perancangan Antarmuka Home Admin



Lampiran 24 Perancangan Antarmuka Olah Data Admin



Lampiran 25 Perancangan Antarmuka Home Surveyor



Lampiran 26 Data Quickcount Pemilihan Walikota Malang Tahun 2013

No.	Kelurahan	dwi-uddin	SR-MK	DADI	RAJA	DOA	AJI	jumlah
1	BLIMBING 1 TPS 1	12	65	88	14	16	184	379
2	BLIMBING 2 TPS1	7	192	29	9	17	115	369
3	BLIMBING 2 TPS2	11	112	21	17	10	137	308
4	BLIMBING 3 TPS 1	25	42	49	8	22	118	264
5	BLIMBING 3 TPS 2	19	37	35	10	15	179	295
6	BLIMBING 3 TPS 3	15	128	41	2	14	153	353
7	BLIMBING 4 TPS 1	20	100	43	3	28	158	352
8	BLIMBING 5 TPS 1	17	65	45	12	13	167	319
9	BLIMBING 5 TPS 2	26	58	35	7	23	180	329
10	BLIMBING 5 TPS 3	6	74	67	12	18	166	343
11	BLIMBING 5 TPS 4	31	64	27	1	7	100	230
12	BLIMBING 6 TPS 1	15	80	47	7	13	169	331
13	BLIMBING 6 TPS 2	13	45	33	12	6	241	350
14	BLIMBING 6 TPS 3	25	55	27	4	18	90	219
15	BLIMBING 6 TPS 4	7	107	51	9	3	59	236
16	BLIMBING 7 TPS 1	24	96	46	2	6	139	313
17	BLIMBING 7 TPS 2	21	68	58	10	13	120	290
18	BLIMBING 7 TPS 3	14	233	20	3	7	70	347
19	BLIMBING 8 TPS 1	44	43	57	9	22	103	278
20	BLIMBING 9 TPS 1	5	169	71	82	10	83	420
21	BLIMBING 9 TPS 2	19	197	53	5	6	105	385
22	BLIMBING 9 TPS 3	13	215	76	7	28	98	437
23	BLIMBING 10 TPS 1	19	40	80	22	5	147	313
24	BLIMBING 10 TPS 2	15	88	36	22	23	126	310
25	BLIMBING 11 TPS 1	17	93	38	11	14	135	308
26	KEDUNGKANDANG 1 TPS 1	5	105	95	0	7	153	365

27	KEDUNGKANDANG 1 TPS 2	5	115	35	11	8	212	386
28	KEDUNGKANDANG 1 TPS 3	6	123	29	0	4	127	289
29	KEDUNG KANDANG 1 TPS 4	5	35	74	0	6	125	245
30	KEDUNGKANDANG 2 TPS 1	9	68	68	9	13	131	298
31	KEDUNGKANDANG 2 TPS 2	16	72	50	4	9	121	272
32	KEDUNGKANDANG 2 TPS 3	13	48	32	5	5	180	283
33	KEDUNGKANDANG 3 TPS 1	9	35	52	9	15	193	313
34	KEDUNGKANDANG 3 TPS 2	16	31	49	3	12	141	252
35	KEDUNGKANDANG 4 TPS 1	2	116	61	2	6	179	366
36	KEDUNGKANDANG 5 TPS 1	25	51	45	9	14	245	389
37	KEDUNGKANDANG 6 TPS 1	8	87	85	1	6	245	432
38	KEDUNGKANDANG 7 TPS 1	13	82	86	5	7	97	290
39	KEDUNGKANDANG 7 TPS 2	19	99	57	1	6	152	334
40	KEDUNGKANDANG 8 TPS 1	20	60	45	11	12	175	323
41	KEDUNGKANDANG 8 TPS 2	13	80	21	4	4	98	220
42	KEDUNGKANDANG 8	24	91	49	4	18	88	274

	TPS 3							
43	KEDUNGKANDANG 8 TPS 4	24	71	26	4	33	87	245
44	KEDUNGKANDANG 9 TPS 1	63	80	72	15	7	118	355
45	KEDUNGKANDANG 9 TPS 2	19	57	33	8	18	121	256
46	KEDUNGKANDANG 9 TPS 3	52	49	44	12	25	110	292
47	KEDUNGKANDANG 10 TPS 1	16	154	43	2	9	74	298
48	KEDUNGKANDANG 10 TPS 2	12	86	62	7	4	195	366
49	KEDUNGKANDANG 11 TPS 1	5	36	64	3	12	127	247
50	KEDUNGKANDANG 12 TPS 1	6	22	92	1	2	232	355
51	KLOJEN 1 TPS 1	28	36	32	3	20	102	221
52	KLOJEN 2 TPS 1	26	71	97	1	14	115	324
53	KLOJEN 3 TPS 1	24	78	79	5	19	118	323
54	KLOJEN 3 TPS 2	17	95	58	5	3	106	284
55	KLOJEN 4 TPS 1	23	52	26	4	4	101	210
56	KLOJEN 5 TPS 1	18	31	43	9	11	249	361
57	KLOJEN 6 TPS 1	16	43	32	10	12	201	314
58	KLOJEN 6 TPS 2	11	50	26	1	5	111	204
59	KLOJEN 7 TPS 1	11	60	34	14	25	228	372
60	KLOJEN 8 TPS 1	12	56	98	3	5	119	293
61	KLOJEN 8 TPS 2	30	55	21	1	1	89	197
62	KLOJEN 9 TPS 1	13	42	64	10	6	166	301
63	KLOJEN 9 TPS 2	10	74	77	1	2	111	275
64	KLOJEN 10 TPS 1	16	94	86	5	11	97	309

65	KLOJEN 11 TPS 1	20	110	44	10	8	79	271
66	LOWOKWARU 1 TPS 1	15	70	53	6	9	225	378
67	LOWOKWARU 2 TPS 1	18	30	12	6	9	212	287
68	LOWOKWARU 2 TPS 2	26	27	19	5	18	230	325
69	LOWOKWARU 3 TPS 1	7	43	13	4	3	173	243
70	LOWOKWARU 3 TPS 2	9	15	12	1	7	281	325
71	LOWOKWARU 4 TPS 1	15	42	50	3	15	200	325
72	LOWOKWARU 4 TPS 2	13	36	46	2	7	207	311
73	LOWOKWARU 5 TPS 1	18	45	36	2	13	216	330
74	LOWOKWARU 6 TPS 1	26	33	29	8	12	196	304
75	LOWOKWARU 7 TPS 1	10	48	26	17	15	189	305
76	LOWOKWARU 7 TPS 2	7	41	34	11	17	194	304
77	LOWOKWARU 7 TPS 3	25	65	24	3	16	137	270
78	LOWOKWARU 8 TPS 1	22	65	52	12	8	151	310
79	LOWOKWARU 8 TPS 2	17	123	34	2	22	124	322
80	LOWOKWARU 9 TPS 1	37	63	27	4	11	151	293
81	LOWOKWARU 9 TPS 2	26	55	23	5	30	68	207
82	LOWOKWARU 9 TPS 3	27	46	30	3	14	120	240
83	LOWOKWARU 10 TPS 1	37	54	26	5	30	154	306
	2	25	55	40	6	6	96	228
85	LOWOKWARU 11 TPS 1	13	56	98	7	7	113	294
	2	30	133	31	4	5	173	376
87	LOWOKWARU 11 TPS 3	25	48	37	5	14	181	310
	1	4	45	61	1	12	298	421
89	SUKUN 1 TPS 1	12	45	52	18	7	227	361

90	SUKUN 1 TPS 2	7	75	63	17	3	215	380
91	SUKUN 2 TPS 1	21	81	54	10	6	169	341
92	SUKUN 2 TPS 2	17	94	58	12	9	111	301
93	SUKUN 2 TPS 3	15	76	67	5	13	157	333
94	SUKUN 3 TPS 1	28	44	113	9	15	120	329
95	SUKUN 4 TPS 1	7	97	133	3	5	111	356
96	SUKUN 4 TPS 2	11	87	82	3	23	114	320
97	SUKUN 4 TPS 3	9	54	99	13	9	71	255
98	SUKUN 4 TPS 4	10	81	74	5	3	214	387
99	SUKUN 5 TPS 1	21	54	67	6	12	230	390
100	SUKUN 5 TPS 2	8	49	84	5	3	293	442
101	SUKUN 5 TPS 3	12	45	137	4	9	158	365
102	SUKUN 6 TPS 1	24	64	55	2	18	148	311
103	SUKUN 6 TPS 2	15	145	46	11	25	170	412
104	SUKUN 6 TPS 3	17	128	73	8	9	163	398
105	SUKUN 6 TPS 4	12	65	41	6	7	209	340
106	SUKUN 7 TPS 1	28	47	25	1	8	95	204
107	SUKUN 7 TPS 2	9	77	26	8	2	95	217
108	SUKUN 8 TPS 1	28	101	46	9	29	164	377
109	SUKUN 8 TPS 2	20	37	70	3	9	120	259
110	SUKUN 9 TPS 1	8	84	71	7	20	194	384
111	SUKUN 9 TPS 2	13	31	108	7	11	152	322
112	SUKUN 10 TPS 1	22	110	75	2	4	117	330
113	SUKUN 10 TPS 2	5	169	71	82	10	83	420
114	SUKUN 11 TPS 1	12	67	234	5	10	84	412
jumlah		1963	8511	6196	895	1344	17133	36042

Lampiran 27 Kasus Uji Validasi Penambahan Data Kecamatan

Nama Kasus Uji	Penambahan Data Kecamatan
Tujuan Pengujian	Untuk menambahkan data kecamatan oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none">1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i>2. <i>Administrator</i> masuk ke halaman data kecamatan dengan menekan tombol data kecamatan3. <i>Administrator</i> masuk ke halaman <i>new record</i> dengan menekan tombol <i>new record</i>4. <i>Administrator</i> memasukkan data kecamatan pada form <i>input</i> yang disediakan, yaitu berupa nama kecamatan blimbing5. <i>Administrator</i> menyimpan data kecamatan dengan menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem dapat menyimpan data kecamatan
Hasil yang didapatkan	Sistem menyimpan dan menampilkan data yang telah tersimpan
Status Validitas	Valid

Lampiran 28 Kasus Uji Validasi Penghapusan Data Kecamatan

Nama Kasus Uji	Penghapusan Data Kecamatan
Tujuan Pengujian	Untuk menghapus data kecamatan oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none">1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i>2. <i>Administrator</i> masuk ke halaman data kecamatan dengan menekan tombol data kecamatan3. <i>Administrator</i> memberi tanda <i>checklist</i> pada kecamatan blimbing yang akan dihapus4. <i>Administrator</i> menekan tombol <i>delete</i>
Hasil yang diharapkan	Sistem dapat menghapus data kecamatan dan menampilkan daftar data yang masih tersimpan
Hasil yang didapatkan	Sistem menghapus data yang telah ditandai dan menampilkan data yang masih tersimpan
Status Validitas	Valid

Lampiran 29 Kasus Uji Validasi Pengubahan Data Kecamatan

Nama Kasus Uji	Pengubahan Data Kecamatan
Tujuan Pengujian	Untuk mengubah data kecamatan oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i> 2. <i>Administrator</i> masuk ke halaman data kecamatan dengan menekan tombol data kecamatan 3. <i>Administrator</i> memberi tanda <i>checklist</i> pada kecamatan blimbing yang akan diubah 4. <i>Administrator</i> menekan tombol <i>edit</i> pada kolom <i>action</i> 5. <i>Administrator</i> mengubah nama kecamatan blimbing menjadi kecamatan lowokwaru kedalam form yang disediakan 6. <i>Administrator</i> menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem dapat mengubah data kecamatan dan menampilkan daftar data yang telah diubah
Hasil yang didapatkan	Sistem menghapus data yang telah ditandai dan menampilkan data yang telah berubah
Status Validitas	Valid

Lampiran 30 Kasus Uji Validasi Penambahan Data Periode

Nama Kasus Uji	Penambahan Data Periode
Tujuan Pengujian	Untuk menambahkan data periode oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i> 2. <i>Administrator</i> masuk ke halaman data periode dengan menekan tombol data periode 3. <i>Administrator</i> masuk ke halaman <i>new record</i> dengan menekan tombol <i>new record</i> 4. <i>Administrator</i> memasukkan data periode pada form input yang disediakan, yaitu berupa “Pemilihan Calon Walikota dan Calon Wakil Walikota Kota Malang periode 2013-2018” 5. <i>Administrator</i> menyimpan data periode dengan menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem dapat menyimpan data periode
Hasil yang didapatkan	Sistem menyimpan dan menampilkan data periode “Pemilihan Calon Walikota dan Calon Wakil Walikota Kota Malang periode 2013-2018” dalam daftar periode
Status Validitas	Valid

Lampiran 31 Kasus Uji Validasi Penghapusan Data Periode

Nama Kasus Uji	Penghapusan Data Periode
Tujuan Pengujian	Untuk menghapus data periode oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i> 2. <i>Administrator</i> masuk ke halaman data periode dengan menekan tombol data periode 3. <i>Administrator</i> memberi tanda <i>checklist</i> pada data periode “Pemilihan Calon Walikota dan Calon Wakil Walikota Kota Malang periode 2013-2018 yang akan dihapus 4. <i>Administrator</i> menekan tombol <i>delete</i>
Hasil yang diharapkan	Sistem dapat menghapus data periode dan menampilkan daftar data yang masih tersimpan
Hasil yang didapatkan	Sistem menghapus data periode “Pemilihan Calon Walikota dan Calon Wakil Walikota Kota Malang periode 2013-2018” dan menampilkan data yang masih tersimpan
Status Validitas	Valid

Lampiran 32 Kasus Uji Validasi Perubahan Data Periode

Nama Kasus Uji	Pengubahan Data Periode
Tujuan Pengujian	Untuk mengubah data periode oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i> 2. <i>Administrator</i> masuk ke halaman data periode dengan menekan tombol data periode 3. <i>Administrator</i> memberi tanda <i>checklist</i> pada data periode “Pemilihan Calon Walikota dan Calon Wakil Walikota Kota Malang periode 2013-2018” yang akan diubah 4. <i>Administrator</i> menekan tombol <i>edit</i> pada kolom <i>action</i> 5. <i>Administrator</i> mengubah data periode baru kedalam form yang disediakan dengan “Pemilihan Calon Walikota & Calon Wakil Walikota Kota Malang periode 2013-2018” 6. <i>Administrator</i> menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem dapat mengubah data periode dan menampilkan daftar data periode yang telah diubah
Hasil yang didapatkan	Sistem mengubah data periode “Pemilihan Calon Walikota & Calon Wakil Walikota Kota Malang periode 2013-2018”
Status Validitas	Valid

Lampiran 33 Kasus Uji Validasi Penambahan Data TPS

Nama Kasus Uji	Penambahan Data TPS
Tujuan Pengujian	Untuk menguji validitas kinerja sistem perangkat lunak dalam menyediakan fasilitas menambahkan data TPS oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i> 2. <i>Administrator</i> masuk ke halaman data TPS dengan menekan tombol data TPS 3. <i>Administrator</i> masuk ke halaman <i>new record</i> 4. <i>Administrator</i> memasukkan data TPS pada form <i>input</i> yang disediakan yaitu memasukkan periode dengan periode “Pemilihan Calon Walikota dan Calon Wakil Walikota Kota Malang periode 2013-2018”, pada kolom kecamatan diisikan kecamatan Blimbing, pada kolom nama diisikan blimbing1 TPS 1, pada kolom alamat diisikan blimbing, dan pada jumlah pemilih tetap diisikan angk 379 5. <i>Administrator</i> menyimpan data TPS dengan menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem dapat menyimpan data TPS
Hasil yang didapatkan	Sistem menyimpan dan menampilkan data TPS blimbing1 TPS 1 dalam daftar data TPS
Status Validitas	Valid

Lampiran 34 Kasus Uji Validasi Penghapusan Data TPS

Nama Kasus Uji	Penghapusan Data TPS
Tujuan Pengujian	Untuk menguji validitas kinerja sistem perangkat lunak dalam menyediakan fasilitas menghapus data TPS oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none">1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i>2. <i>Administrator</i> masuk ke halaman data TPS dengan menekan tombol data TPS3. <i>Administrator</i> memberi tanda <i>checklist</i> pada data TPS blimbing1 TPS 1 yang akan dihapus4. <i>Administrator</i> menekan tombol <i>delete</i>
Hasil yang diharapkan	Sistem dapat menghapus data TPS dan menampilkan daftar data TPS yang masih tersimpan
Hasil yang didapatkan	Sistem menghapus data TPS blimbing1 TPS1
Status Validitas	Valid

Lampiran 35 Kasus Uji Validasi Perubahan Data TPS

Nama Kasus Uji	Pengubahan Data TPS
Tujuan Pengujian	Untuk mengubah data TPS oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none">1. <i>Administrator</i> melakukan login dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i>2. <i>Administrator</i> masuk ke halaman data TPS dengan menekan tombol data TPS3. <i>Administrator</i> memberi tanda <i>checklist</i> pada data TPS blimbing 1 TPS 1 yang akan diubah4. <i>Administrator</i> menekan tombol <i>edit</i> pada kolom <i>action</i>5. <i>Administrator</i> mengubah data TPS “Kec blimbing1 TPS 1” kedalam form yang disediakan6. <i>Administrator</i> menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem dapat mengubah data TPS dan menampilkan daftar data TPS yang telah diubah
Hasil yang didapatkan	Sistem mengubah data TPS TPS blimbing 1 TPS 1 dan menampilkan data TPS yang telah berubah menjadi “Kec blimbing1 TPS 1”
Status Validitas	Valid

Lampiran 36 Kasus Uji Validasi Penambahan Data Kandidat

Nama Kasus Uji	Penambahan Data Kandidat
Tujuan Pengujian	Untuk menambahkan data kandidat oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i> 2. <i>Administrator</i> masuk ke halaman data kandidat dengan menekan tombol data kandidat 3. <i>Administrator</i> masuk ke halaman <i>new record</i> 4. <i>Administrator</i> memasukkan data kandidat yang berupa nama kandidat yang diisi dengan nama kandidat 1 “Moch. Anton”, nama kandidat 2 diisi dengan “Sutiaji”, partai pendukung diisi dengan “PKB dan Gerindra”, visi dan misi diisi dengan “Menjadi walikota merupakan suatu power untuk menyejahterakan rakyat.”, nomor urut diisi angka 4, foto diisi dengan foto pasangan, dan periode diisi pilihan periode 2013-2018 pada form <i>input</i> yang disediakan. 5. <i>Administrator</i> menyimpan data kandidat dengan menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem dapat menyimpan data kandidat
Hasil yang didapatkan	Sistem menyimpan dan menampilkan data kandidat nomor urut 4 pada daftar data kandidat yang telah tersimpan
Status Validitas	Valid

Lampiran 37 Kasus Uji Validasi Penghapusan Data Kandidat

Nama Kasus Uji	Penghapusan Data Kandidat
Tujuan Pengujian	Untuk menghapus data kandidat oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none">1. <i>Administrator</i> melakukan login dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i>2. <i>Administrator</i> masuk ke halaman data kandidat dengan menekan tombol data kandidat3. <i>Administrator</i> memberi tanda <i>checklist</i> pada data kandidat nomor urut 1 yang akan dihapus4. <i>Administrator</i> menekan tombol <i>delete</i>
Hasil yang diharapkan	Sistem dapat menghapus data kandidat dan menampilkan daftar data kandidat yang masih tersimpan
Hasil yang didapatkan	Sistem menghapus data kandidat nomor urut 1 dan menampilkan data kandidat yang masih tersimpan
Status Validitas	Valid

Lampiran 38 Kasus Uji Validasi Pengubahan Data Kandidat

Nama Kasus Uji	Pengubahan Data Kandidat
Tujuan Pengujian	Untuk mengubah data kandidat oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none">1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i>2. <i>Administrator</i> masuk ke halaman data kandidat dengan menekan tombol data kandidat3. <i>Administrator</i> memberi tanda <i>checklist</i> pada data kandidat nomor urut 1 yang akan diubah4. <i>Administrator</i> menekan tombol <i>edit</i> pada kolom <i>action</i>5. <i>Administrator</i> mengubah data kandidat baru kedalam form yang disediakan, berupa merubah nama kandidat nomor urut 1 menjadi Dwi & Udin6. <i>Administrator</i> menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem dapat mengubah data kandidat dan menampilkan daftar data kandidat yang telah diubah
Hasil yang didapatkan	Sistem mengubah data kandidat nomor urut 1 dan menampilkan data kandidat yang telah berubah menjadi kandidat nomor urut 1 menjadi Dwi & Udin
Status Validitas	Valid

Lampiran 39 Kasus Uji Validasi Penambahan Data User

Nama Kasus Uji	Penambahan Data <i>User</i>
Tujuan Pengujian	Untuk menambahkan data <i>user</i> oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i> 2. <i>Administrator</i> masuk ke halaman data <i>user</i> dengan menekan tombol data <i>user</i> 3. <i>Administrator</i> masuk ke halaman <i>new record</i> 4. <i>Administrator</i> memasukkan data <i>user</i> yang berupa <i>username</i> yang diisi dengan server, nama yang diisi dengan surveyor1, password diisi dengan server, alamat diisi dengan Malang, nomor telepon diisi dengan 123456, memilih status menjadi surveyor, mengisi tanggal dengan tanggal terdaftar pada form <i>input</i> yang disediakan. 5. <i>Administrator</i> menyimpan data <i>user</i> dengan menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem dapat menyimpan data <i>user</i>
Hasil yang didapatkan	Sistem menyimpan dan menampilkan data <i>user</i> pada daftar data <i>user</i> yang telah tersimpan
Status Validitas	Valid

Lampiran 40 Kasus Uji Validasi Penghapusan Data User

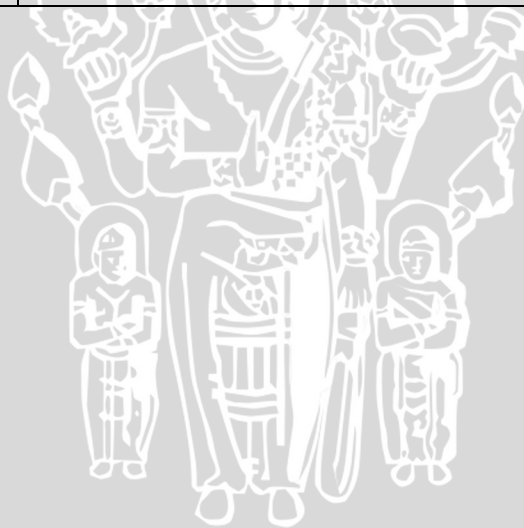
Nama Kasus Uji	Penghapusan Data <i>User</i>
Tujuan Pengujian	Untuk menghapus data <i>user</i> oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none">1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i>2. <i>Administrator</i> masuk ke halaman data <i>user</i> dengan menekan tombol data <i>user</i>3. <i>Administrator</i> memberi tanda <i>checklist</i> pada data <i>user</i> server yang akan dihapus4. <i>Administrator</i> menekan tombol <i>delete</i>
Hasil yang diharapkan	Sistem dapat menghapus data <i>user</i> dan menampilkan daftar data <i>user</i> yang masih tersimpan
Hasil yang didapatkan	Sistem menghapus data <i>user</i> server pada daftar <i>user</i> dan menampilkan data <i>user</i> yang masih tersimpan
Status Validitas	Valid

Lampiran 41 Kasus Uji Validasi Pengubahan Data User

Nama Kasus Uji	Pengubahan Data <i>User</i>
Tujuan Pengujian	Untuk mengubah data <i>user</i> oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Administrator</i> melakukan <i>login</i> dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i> 2. <i>Administrator</i> masuk ke halaman data <i>user</i> dengan menekan tombol data <i>user</i> 3. <i>Administrator</i> memberi tanda <i>checklist</i> pada data <i>user</i> yaitu server yang akan diubah 4. <i>Administrator</i> menekan tombol <i>edit</i> pada kolom <i>action</i> 5. <i>Administrator</i> mengubah data baru kedalam form yang disediakan, mengubah <i>password</i> server menjadi survey1 6. <i>Administrator</i> menekan tombol <i>save</i>
Hasil yang diharapkan	Sistem dapat mengubah data <i>surveyor</i> dan menampilkan daftar data <i>surveyor</i> yang telah diubah
Hasil yang didapatkan	Sistem mengubah data server yang telah ditandai dan menampilkan data server dengan <i>password</i> yang telah berubah server menjadi survey1
Status Validitas	Valid

Lampiran 42 Kasus Uji Validasi Menampilkan Data Log Quickcount

Nama Kasus Uji	Menampilkan Data <i>Log Quickcount</i>
Tujuan Pengujian	Untuk menampilkan data <i>log quickcount</i> oleh <i>administrator</i>
Prosedur Uji	<ol style="list-style-type: none">1. <i>Administrator</i> melakukan login dengan mengisi kolom <i>username</i> dengan kata admin, kolom <i>password</i> dengan kata admin, dan menekan tombol <i>login</i>2. <i>Administrator</i> masuk ke halaman data <i>Log Quickcount</i> dengan menekan tombol <i>log quickcount</i>
Hasil yang diharapkan	Administrator dapat melihat data <i>log quickcount</i>
Hasil yang didapatkan	Sistem menampilkan data <i>log quickcount</i>
Status Validitas	Valid



Lampiran 43 Kuisisioner Pengguna

