



PENGEMBANGAN APLIKASI

CAPI

(COMPUTER ASSISTED PERSONAL INTERVIEW)

DINAMIS

Berbasis Android



STUDI

PARIWISATA BERKELANJUTAN

**DI KABUPATEN LOMBOK BARAT DAN LOMBOK UTARA
TAHUN 2016**

**PRAKTIK KERJA LAPANGAN
TAHUN AKADEMIK 2015/2016**



PRAKATA

Praktik Kerja Lapangan (PKL) merupakan bagian dari kurikulum Program Diploma IV Sekolah Tinggi Ilmu Statistik (STIS) yang bertujuan untuk melatih mahasiswa STIS dalam melaksanakan penelitian lapangan secara ilmiah. PKL digunakan sebagai sarana bagi mahasiswa tingkat III STIS untuk menerapkan ilmu yang diperoleh selama berkuliah di STIS. Dalam pelaksanaan PKL STIS Tahun Akademik 2015/2016 dilakukan sebuah penelitian terkait metode pengumpulan data yang memanfaatkan implementasi teknologi informasi. Hasil dari penelitian tersebut kemudian disusun dalam sebuah publikasi yang berjudul ***“Pengembangan Aplikasi Computer Assisted Personal Interviewing (CAPI) Berbasis Android”***. Publikasi ini disusun berdasarkan penggunaan *Computer Assisted Personal Interviewing (CAPI)* pada kegiatan pengumpulan data di lapangan pada PKL 2015/2016 di dua tempat, yaitu Kabupaten Lombok Barat dan Kabupaten Lombok Utara.

Semoga publikasi hasil penelitian PKL STIS Tahun Akademik 2015/2016 ini dapat bermanfaat sebagai *prototipe* pengembangan aplikasi CAPI di Badan Pusat Statistik (BPS) Republik Indonesia, lingkungan akademik STIS, dan pihak lain yang terkait dalam upaya pengembangan metode pengumpulan data yang lebih optimal dan canggih. Akhirnya kami mengucapkan terima kasih sebesar-besarnya kepada seluruh pihak yang telah membantu selama proses kegiatan ini berlangsung.

Jakarta, Desember 2016
Praktik Kerja Lapangan 2015/2016
Ketua Penyelenggara,

Ir. Ekaria, M.Si
NIP 19620722 198501 2 001



ABSTRACT

PKL STIS 2016, “Application Development of Dynamis CAPI with Android Based”

vii + 80 pages

In practice, the survey or census generally using an enumeration method, that is the method of PAPI (Paper and Pencil Interviewing). Where in this method the enumerators are required to perform the respondents data recording using questionnaire and a pencil tool. However, this method is less effective and efficient because it requires the questionnaire to entry again into the computer in order to be easily processed raw data, but it also increases the chance of human error due to go through the entry process again. Therefore we need a CAPI (Computer Assisted Personal Interviewing) method that can directly record the data with optimal validation and do not need to go through the process of batching, editing, coding, and data entry.

On the activities of PKL 54, CAPI method has been applied into the directly enumeration process in the field. However, deficiencies are still found on CAPI information system development process is due to the questionnaire that has not been fixed which forces developers to wait a questionnaire that has been fixed. In addition, if the information system was ready for use and it still happens a change in the questionnaire, then the developer is required to unlock the code and fix it raw. Of course it is extremely inefficient, especially when a lot of changes in the questionnaire. Therefore, at this time PKL 55 built a new system based modular CAPI where the questionnaire is not the built-in at the CAPI, but separate from the system so no need to change the questionnaire involves changing the code. In this dynamic CAPI expected to be able to cover the shortfall that occurred in CAPI PKL 54.

Therefore, in this study will be assessed on the development of information systems of dynamic CAPI, its implementation in the field covering the efficiency and effectiveness of the enumeration, the reaction enumerators and respondents in





the field, and also advantages and disadvantages possessed by dynamic CAPI PKL 55 PKL 54 compared with CAPI and PAPI use at PKL 55.

In the development of the system CAPI PKL 55 using the ODK Aggregate providing server ready to deploy. While research method used is to use an experimental method in which the research experiment conducted directly in the field at 55. Where on the activities of the PKL 55 at the implementation is done using PAPI and CAPI once with each proportion is 50%.

The results of this study show success in development of information systems of dynamic CAPI that is able to cover the shortfall PKL 54 CAPI. In addition, the CAPI implementation proved to be more efficient and effective when compared to using PAPI. However, they found a weakness that appears on its implementation, namely the availability of search devices and system software that sufficient to be able to run this information system of dynamic CAPI optimally. Thus, the next research / PKL expected CAPI to grow further so as to realize the optimal enumeration in terms of quality, cost, and time.

Keywords: Enumeration, CAPI, PAPI, PKL.



DAFTAR ISI

	Halaman
PRAKATA	i
ABSTRAK.....	ii
DAFTAR ISI	iv
DAFTAR TABEL	v
DAFTAR GAMBAR.....	vi
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Identifikasi dan Batasan Masalah.....	3
1.3 Rumusan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
BAB II KAJIAN PUSTAKA	
2.1 Landasan Teori	7
2.2 Penelitian Terkait.....	18
BAB III PEMBANGUNAN SISTEM	
3.1 <i>Build</i>	21
3.2 <i>Aggregate</i>	34
3.3 Pengumpulan Data (<i>Collect</i>)	39
3.4 <i>Sampling Management</i>	59
BAB IV IMPLEMENTASI SISTEM (<i>Standard Operational Procedure</i>)	
4.1 Deskripsi Penggunaan CAPI	73
4.2 Profil Pengguna CAPI	74
4.3 Penjelasan per Fungsi	75
BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan.....	79
5.2 Saran	79





DAFTAR TABEL

No. Tabel	Nama Tabel	Halaman
1	Lima kategori skala likert.....	14
2	Tipe pertanyaan dalam pembuatan kuesioner CAPI	26
3	Tabel pengaturan validasi dalam pembuatan kuesioner CAPI.....	28
4	Tabel penjelasan <i>worksheet survey</i> pada <i>.xls form</i>	30
5	Fungsi validasi pada kuesioner CAPI	32





DAFTAR GAMBAR

No. Gambar	Nama Gambar	Halaman
1	Alur kerja subsistem <i>build</i> kuesioner aplikasi CAPI.....	21
2	<i>Data flow diagram</i> subsistem <i>build</i> CAPI 55.....	23
3	<i>Entity relationship diagram</i> subsistem <i>build</i> aplikasi CAPI	23
4	<i>Use case diagram</i> pada subsistem <i>build</i> aplikasi CAPI 55	24
5	Halaman pertama <i>form</i> pembuatan kuesioner	25
6	Tampilan tipe pertanyaan pada pembuatan kuesioner CAPI.....	26
7	Pengaturan pertanyaan dalam pembuatan kuesioner CAPI.....	27
8	Pengaturan validasi dalam pembuatan kuesioner CAPI	27
9	Tampilan <i>worksheet survey</i> pada <i>.xls form</i>	29
10	Tampilan <i>sheet “choices”</i> pada <i>.xls form</i>	31
11	Tampilan <i>Upload Submission Data</i>	38
12	Tampilan <i>Manually Upload Submission Data</i>	38
13	Tampilan apabila kuesioner sukses diunggah.....	39
14	Alur kerja aplikasi CAPI.....	39
15	<i>Bussines Process</i> pada proses <i>Login</i>	41
16	<i>Bussines Process</i> pada proses <i>Download</i> Kuesioner	42
17	<i>Bussines Process</i> pada proses Menghapus Data Kuesioner	43
18	<i>Data Flow Diagram</i> pada aplikasi CAPI.....	44
19	<i>Activity Diagram</i> pada proses Pengisian Kuesioner	45
20	<i>Activity Diagram</i> pada proses Penampilan Hasil atau Wawancara	45
21	<i>Activity Diagram</i> pada proses Pengisian Wawancara.....	46
22	<i>Activity Diagram</i> saat mengunduh isian kuesioner.....	47
23	<i>Activity Diagram</i> proses Notifikasi Perubahan Status.....	48
24	<i>Entity Relationship Diagram</i> CAPI PKL 55.....	49
25	<i>Use Case</i> dalam Aplikasi CAPI.....	50
26	Tampilan depan aplikasi CAPI.....	51
27	Tampilan aplikasi CAPI setelah pengguna <i>Login</i>	51
28	Tampilan menu isi kuesioner pada aplikasi CAPI.....	52
29	Tampilan fungsi-fungsi pada aplikasi CAPI.....	53





30	Tampilan menu Hapus Data pada aplikasi CAPI.....	53
31	Tampilan Ubah Hasil Wawancara pada aplikasi CAPI	54
32	Tampilan <i>Download</i> Kuesioner pada aplikasi CAPI	54
33	Tampilan Kirim Hasil Wawancara pada aplikasi CAPI.....	55
34	Tampilan Monitoring pada aplikasi CAPI	56
35	Tampilan <i>Sampling Management</i> pada aplikasi CAPI.....	56
36	Tampilan dalam Proses <i>Sampling</i> pada aplikasi CAPI	57
37	Tampilan dalam proses konfirmasi kortim pada aplikasi CAPI	58
38	Tampilan cara mengunduh buku pedoman pada aplikasi CAPI	58
39	Tampilan pengaturan pada aplikasi CAPI.....	59
40	<i>Flowchart</i> Desain <i>Sampling</i> untuk populasi rumah tangga.....	60
41	<i>Data Flow Diagram Sampling Management</i>	63
42	<i>Entity Relationship Diagram Sampling Management</i>	64
43	<i>Use Case</i> pada <i>Sampling Management</i>	65
44	<i>Flowchart</i> Pemilihan Blok Sensus pada <i>Sampling Management</i>	66
45	<i>Flowchart</i> dalam pengisian Keterangan Rumah Tangga	67
46	<i>Flowchart</i> dalam Proses Listing.....	68
47	<i>Flowchart</i> Penarikan Sampel pada <i>Sampling Management</i>	68
48	<i>Flowchart</i> Penyimpanan Hasil Penarikan Sampel	69
49	<i>Flowchart</i> dalam proses <i>Upload</i> Hasil <i>Sampling</i>	70
50	<i>Flowchart</i> dalam Menampilkan Hasil Listing	71
51	Alur Pencacahan menggunakan CAPI	73
52	Tampilan awal <i>login</i> pada Aplikasi CAPI.....	75





Halaman ini sengaja dikosongkan





BAB I

PENDAHULUAN

1.1 Latar Belakang

Badan Pusat Statistik (BPS) merupakan Lembaga Pemerintah Non Kementerian yang memiliki peran sebagai penyedia kebutuhan data bagi pemerintah dan masyarakat. Data dan informasi statistik yang dihasilkan BPS digunakan sebagai bahan rujukan untuk menyusun perencanaan, merumuskan kebijakan, dan melakukan evaluasi.

Tuntutan masyarakat terhadap ketersediaan data dan informasi statistik yang beragam dan berkualitas semakin hari semakin meningkat. Pengguna data menginginkan data tersedia lebih cepat, lebih murah, lebih mudah dan lebih berkualitas.

Selama ini data dikumpulkan oleh koordinator statistik kecamatan (KSK) atau pencacah lain dengan melakukan wawancara yang bersifat langsung (*face to face*), kemudian jawaban responden diisikan pada media kuesioner kertas. Metode ini sering disebut dengan metode *Paper and Pencil Interviewing* (PAPI). Teknik wawancara dengan metode PAPI masih memiliki beberapa kelemahan. *The National Research Council U.S* mengklasifikasikan kelemahan tersebut ke dalam beberapa poin penting, di antaranya:

1. Hasil isian harus dimasukkan lagi secara manual ke dalam komputer sehingga membutuhkan waktu dan biaya,
2. Kemungkinan terjadi interpretasi informasi secara ganda meningkat karena tidak ada standardisasi dalam penulisan isian oleh pencacah,





3. Berpeluang besar untuk terjadi kesalahan ketik pada saat entri data.

Untuk mengatasi kelemahan-kelemahan tersebut, dikembangkan teknik wawancara dengan teknologi komputer atau disebut dengan istilah *Computer Assisted Personal Interviewing* (CAPI). CAPI ialah metode pencacahan responden dengan bantuan teknologi komputer, biasanya dengan *notebook*, *laptop*, *PC tablet*, atau perangkat lain yang bisa digunakan. Pada survey yang menggunakan metode CAPI, pengumpulan data dibantu oleh sebuah program *data entry* yang sudah terpasang dalam perangkat yang akan digunakan dan kemudian perangkat tersebut dibawa ke lapangan untuk diisikan langsung dengan data responden atau dapat disebut sebagai entri langsung di lapangan.

Tujuan dari penggunaan CAPI ini adalah mempercepat perekaman, pengumpulan dan pengolahan data serta diharapkan mampu meningkatkan akurasi karena data terkoreksi langsung di lapangan pada saat diisikan. Tujuan lainnya adalah untuk membandingkan metode ini dengan metode PAPI dari berbagai aspek.

Pemanfaatan CAPI diharapkan dapat meminimalisir biaya yang dikeluarkan akibat penggunaan kertas, pensil, penghapus dan alat-alat lainnya yang digunakan jika menggunakan metode PAPI. Dari segi kesalahan pencacah, penggunaan metode CAPI diharapkan juga mampu melakukan validasi secara langsung saat pencacah CAPI mengisikan *field* yang ada pada kuesioner dan kemudian sejumlah pesan kesalahan akan keluar yang meminta agar pencacah segera mengecek kembali data yang diisikan jika aplikasi menemukan kesalahan.

Waktu yang diperlukan sejak data didapatkan hingga diolah menjadi lebih singkat karena data yang ada akan tersimpan dalam bentuk *softcopy* yang





kemudian akan segera digabungkan dengan data dari perangkat lain. Tidak adanya proses pengentrian data secara massal setelah proses pengumpulan dan pengolahan menjadikan penyajian hasil analisis data juga dapat lebih cepat.

Praktik Kerja Lapangan 54 Sekolah Tinggi Ilmu Statistik (PKL 54 STIS) telah menerapkan metode CAPI untuk mengumpulkan data. Namun, berbeda dengan CAPI PKL 54 yang merupakan CAPI statis, CAPI yang digunakan oleh PKL 55 merupakan CAPI yang bersifat dinamis sehingga tidak hanya dapat digunakan untuk satu form kuisioner pencacahan saja. Dengan pengembangan yang dilakukan pada CAPI pada PKL 55 STIS ini, diharapkan aplikasi ini dapat digunakan secara lebih luas dan tidak terbatas hanya untuk PKL 55 saja.

1.2 Identifikasi dan Batasan Masalah

Penggunaan metode PAPI dan CAPI dalam berbagai survei, masih memiliki berberapa permasalahan dalam pelaksanaannya. Beberapa permasalahan tersebut adalah:

1. Waktu yang diperlukan untuk melakukan wawancara dengan metode PAPI lebih lama karena pencacah harus melakukan validasi setiap pertanyaan secara manual,
2. Dengan metode PAPI sering ditemukan isian yang tidak lengkap, hal ini karena validasi dilakukan secara manual sehingga terkadang ada pertanyaan yang terlewat,
3. Biaya untuk mencetak kuesioner kertas dan peralatan lainnya yang diperlukan untuk melakukan pencacahan dengan metode PAPI cukup besar,



4. Hasil pencacahan dengan metode PAPI harus melalui proses *batching*, *editing*, dan *coding*. Selain itu harus dilakukan input secara manual hasil pencacahan tersebut ke dalam komputer. Tentu saja hal ini membutuhkan waktu dan biaya yang cukup besar dalam proses pengolahan,
5. Metode CAPI yang pernah dilaksanakan pada PKL 54 masih menggunakan CAPI statis, sehingga terbatas untuk form kuesioner tertentu saja.

1.3 Rumusan Masalah

Berdasarkan permasalahan-permasalahan yang telah disebutkan sebelumnya, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana proses pembuatan aplikasi CAPI untuk pencacahan pada PKL 55 STIS?
2. Bagaimana kuantitas waktu pencacahan dengan menggunakan metode CAPI dibandingkan dengan metode PAPI pada PKL 55 STIS?
3. Bagaimana reaksi pencacah dalam melakukan pencacahan dengan menggunakan metode CAPI pada PKL 55 STIS?
4. Apa saja kelebihan dan kelemahan penggunaan metode CAPI dan PAPI pada PKL 55 STIS?

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mengetahui proses pembuatan aplikasi CAPI untuk pencacahan pada PKL 55 STIS,





2. Untuk mengidentifikasi kelebihan dan kelemahan penggunaan metode CAPI dan PAPI pada PKL 55 STIS,
3. Untuk mengidentifikasi kelebihan dan kelemahan penggunaan metode CAPI statis seperti pada PKL 54 STIS dan dinamis pada PKL 55 STIS.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat antara lain:

1. Bagi BPS, sebagai gambaran, bahan masukan dan pertimbangan terkait dengan penggunaan kuesioner berbasis teknologi dalam pengumpulan data untuk survei-survei selanjutnya,
2. Sebagai bahan evaluasi CAPI yang telah dibuat pada PKL 55 STIS yang akan bermanfaat untuk PKL selanjutnya,
3. Sebagai bahan pustaka atau referensi dan perbandingan untuk penelitian selanjutnya.



Halaman ini sengaja dikosongkan





BAB II

KAJIAN PUSTAKA

2.1 Landasan Teori

Praktik Kerja Lapangan Tahun Akademik 2015/2016

Sekolah Tinggi Ilmu Statistik adalah sekolah tinggi kedinasan yang menyiapkan insan statistik untuk menjadi pegawai Badan Pusat Statistik (BPS) di seluruh Indonesia. Sebagai calon pegawai BPS, setiap mahasiswa STIS harus mampu melaksanakan kegiatan utama BPS dalam mengumpulkan data-data di lapangan dan menyusunnya dalam bentuk laporan. Oleh karena itu, pada tingkat III dilakukan kegiatan akademik Praktik Kerja Lapangan (PKL) untuk memahami lebih dalam bagaimana kegiatan utama BPS pada mahasiswa STIS. Kegiatan ini juga bertujuan untuk menerapkan ilmu pengetahuan yang telah dipelajari selama kuliah di STIS sebagai perwujudan Tri Darma Perguruan Tinggi.

Dalam Praktik Kerja Lapangan Tahun Akademik 2015/2016, populasi penelitian dibedakan menjadi tiga, yaitu rumah tangga, unit usaha, dan wisatawan. Topik yang diangkat dalam PKL 55 ini adalah tentang pariwisata berkelanjutan dengan judul “Studi Pariwisata Berkelanjutan Di Kabupaten Lombok Barat dan Lombok Utara Tahun 2016”.

Adapun tujuan penelitian di antaranya adalah:

1. Menggambarkan profil wisatawan, unit usaha pariwisata dan masyarakat ditinjau dari dimensi ekonomi, sosial dan lingkungan di Lombok Barat dan Lombok Utara.





2. Menganalisis indikator-indikator pariwisata berkelanjutan di Lombok Barat dan Lombok Utara.

Data dalam penelitian ini bersumber dari data primer didapatkan langsung dari turun ke lapangan dari wilayah sampel terpilih di Kabupaten Lombok Barat dan Lombok Utara.

Computer Assisted Personal Interviewing (CAPI)

Dari *Social Research* yang dilakukan oleh Roy Sainsbury, John Ditch dan Sandra Hutton mengenai *Computer Assisted Personal Interviewing (CAPI)* (2005), dijelaskan bahwa CAPI merupakan ide yang sederhana dengan dalih daripada mengumpulkan data dengan kuesioner kertas, pewawancara dapat menggunakan komputer portabel untuk memasukkan data secara langsung melalui *keyboard*. CAPI mengklaim mampu meningkatkan kualitas data dengan cara:

1. Menghilangkan masalah *routing* dalam kuesioner,
2. Pewawancara tidak dapat melewati pertanyaan atau mengajukan pertanyaan yang salah,
3. Pertanyaan dapat “disesuaikan” dengan benar,
4. Perhitungan matematis dapat dilakukan dalam program ini,
5. Pemeriksaan secara komputerisasi untuk respon yang tidak konsisten, dan
6. Menghilangkan kesalahan (*errors*) dari proses entri data yang dilakukan secara terpisah.

CAPI dapat menangani *routing*, dimana program akan secara otomatis pindah ke pertanyaan selanjutnya jika pertanyaan sebelumnya sudah terisi dengan benar. Ini merupakan hal yang sangat penting jika suatu pertanyaan tidak





seederhana (kompleks) seperti perulangan dan kondisi-kondisi tertentu yang harus dipenuhi oleh setiap pertanyaan. Selain itu, CAPI dapat melakukan penghitungan matematik, sehingga data yang disimpan juga dapat berupa hasil penghitungan.

CAPI menerapkan aturan-aturan (*rules*) pada pertanyaan yang dibuat sehingga CAPI tidak dapat menerima respon yang tidak konsisten dengan pertanyaan yang diajukan. Jika suatu respon tidak konsisten dengan pertanyaan, maka CAPI dapat menampilkan pesan *error* kepada pewawancara maupun responden. Pewawancara CAPI memasukkan data secara langsung ke komputer sehingga proses entri data yang biasanya dilakukan pada kuesioner kertas tidak lagi diperlukan. Ini dapat menghilangkan salah satu sumber kesalahan saat meng-*entry* dari kuesioner kertas ke komputer dan menghemat waktu serta biaya.

Evaluasi Penggunaan CAPI

Untuk mengetahui gambaran penerimaan dan evaluasi sistem aplikasi CAPI oleh pencacah PKL 54 STIS, telah dilakukan penelitian *Technology Acceptance Modelling* (TAM) dan Evaluasi Sistem oleh tim dosen STIS dengan menggunakan dua kuesioner yang berbeda. Publikasi ini mengacu pada kuesioner dari kedua penelitian tersebut yang memiliki indikator-indikator masing-masing. Sedangkan jenis skala yang digunakan oleh kuesioner tersebut adalah skala likert.

Indikator Penerimaan Sistem (TAM)

Berdasarkan jurnal Model Perilaku Penggunaan IT Pengembangan dari TAM (2007), *Technology Acceptance Model* adalah suatu teori pengembangan





penerimaan sistem berasal dari sebuah teori yang diformulasikan oleh Fishbein & Ajzen (1975) dengan nama *Theory of Reasoned Action* (TRA). Pengembangan lanjut dari TRA adalah *Theory of Planned Behaviour*. Teori ini mengandung beberapa variabel yang akan diadopsi dalam TAM dan digunakan dalam penelitian ini, di antaranya adalah *Subjective Norm*, *Output Quality*, *Perceive Usefulness* dan *Perceive Ease of Use*. Berikut ini penjelasan dari variabel tersebut.

1. *Subjective Norm*

Sejauh mana seseorang memiliki motivasi untuk mengikuti pandangan orang terhadap perilaku yang akan dilakukannya (*Normative Belief*). Jika individu merasa itu adalah hak pribadinya untuk menentukan apa yang akan dilakukan dan tidak ditentukan oleh orang lain di sekitarnya, dia akan mengabaikan pandangan orang tentang perilaku yang akan dilakukannya. Fishbein & Ajzen (1975) menggunakan istilah *motivation to comply* untuk menggambarkan fenomena ini, yaitu apakah individu mematuhi pandangan orang lain yang berpengaruh dalam hidupnya atau tidak.

Berikut merupakan indikator *subjective norm* menurut Lee & Wan (2010) :

- a. Perilaku teman-teman di sekitar akan mempengaruhi dalam penerimaan dalam penggunaan teknologi informasi,
- b. Orang-orang yang sudah berpengalaman dalam menggunakan teknologi informasi dapat mempengaruhi dalam penerimaan penggunaan teknologi informasi.





2. *Output Quality*

ialah derajat dimana seseorang percaya bahwa menggunakan sistem meningkatkan penyelesaian tugasnya dengan baik.

3. *Perceived Usefulness*

David (1989) serta Adams, Nelson, dan Todd (1992) mendefinisikan persepsi kemanfaatan (*perceived usefulness*) merupakan tingkatan kepercayaan seseorang terhadap penggunaan suatu subyek tertentu yang dapat memberikan manfaat bagi orang yang menggunakannya. Chin dan Todd (1995), kemanfaatan dapat terbagi menjadi dua kategori, yaitu:

a. *Usefulness* dengan estimasi satu faktor yang meliputi dimensi:

- Menjadikan pekerjaan lebih mudah (*makes job easier*),
- Bermanfaat (*useful*),
- Menambah produktivitas (*increase productivity*),
- Mempertinggi efektivitas (*enhance effectiveness*),
- Mengembangkan kinerja pekerjaan (*improve job performance*).

b. *Usefulness* dengan estimasi dua faktor, yang meliputi dimensi:

- Menjadikan pekerjaan lebih mudah (*makes job easier*),
- Bermanfaat (*useful*),
- Menambah produktivitas (*increase productivity*).

4. *Perceived Ease of Use*

David (1989) mendefinisikan persepsi kemudahan penggunaan (*perceived ease of use*) merupakan suatu tingkatan kepercayaan seseorang bahwa komputer dapat dengan mudah dipahami. Menurut Adams, Nelson, & Todd (1992) intensitas penggunaan dan interaksi antara pengguna dengan





sistem juga dapat menunjukkan kemudahan penggunaan. Sistem yang lebih sering digunakan menunjukkan bahwa sistem tersebut lebih mudah dipahami, lebih mudah dioperasikan dan lebih mudah untuk digunakan. Berdasarkan definisi tersebut maka dapat disimpulkan bahwa kemudahan penggunaan komputer tergantung pada tingkat kepercayaan seseorang bahwa komputer tersebut dapat dengan mudah dipahami dan sistem yang digunakan dapat dengan mudah dipahami, dioperasikan, dan digunakan.

Lee & Wan (2010) menjelaskan beberapa indikator *perceived ease of use*, antara lain meliputi:

- a. Teknologi informasi sangat mudah dipelajari,
- b. Mudah terampil dalam penggunaan teknologi informasi, dan
- c. Teknologi informasi sangat mudah untuk dioperasikan

Indikator Evaluasi Sistem

1. *System Usability*

Hassan Sadaeghi Naeini dan Sara Mostowfi (2015) dalam penelitian *Using QUIS as a Measurement Tool for User Satisfaction Evaluation (Case Study: Vending machine)* menyebutkan bahwa terdapat dua tipe pengukuran *Usability* suatu sistem maupun software yang bertujuan untuk mendapatkan pemahaman yang lebih lengkap dari kebutuhan pengguna, yaitu *usability in use* (pengukuran hasil penggunaan sistem secara menyeluruh) dan *interface quality* (pengukuran kualitas dari tatap muka sebuah sistem).

Berikut ini merupakan pendefinisian *usability* menurut ISO 9241 -11:

- a. *Effetiveness*: Keakuratan dan kelengkapan.





- b. *Efficiency*: Seberapa cepat pengguna dapat melakukan pekerjaannya dengan penggunaan sistem terkait.
- c. *Satisfaction*: Sejauh mana harapan pengguna terpenuhi.
- d. *Context conformity*: Sejauh mana *usability in use* memenuhi persyaratan dalam semua konteks yang diperlukan penggunaan.
- e. *Context conformity in use*: Capaian tingkat kegunaan sistem secara menyeluruh terhadap persyaratan *usability in use* yang telah ditentukan.
- f. *Context extendibility in use*: Tingkat kegunaan sistem secara menyeluruh di luar konteks persyaratan *usability in use* yang telah ditentukan.
- g. *Accessibility in use*: Tingkat penggunaan bagi pengguna dengan ketidakmampuan tertentu.
- h. *Safety*: Tingkat resiko yang dapat diterima terkait bahaya terhadap orang, bisnis, data, perangkat lunak.

2. Penskalaan

Menurut Steven (1959), penskalaan adalah suatu teknik bagaimana mendapatkan angka yang memberikan arti untuk menilai suatu atribut berdasarkan aturan tertentu. Jenis skala lain yang digunakan untuk mengukur sikap dan respons seseorang terhadap suatu objek adalah skala likert. Skala ini digunakan untuk menguji seberapa kuat derajat setuju atau tidak setuju responden terhadap suatu pertanyaan. Skala ini berwujud kumpulan pertanyaan-pertanyaan sikap yang ditulis, disusun dan dianalisis sedemikian rupa sehingga respons seseorang terhadap pertanyaan tersebut dapat diberikan skor dan kemudian dapat diinterpretasikan. Skala likert tidak terdiri dari



hanya satu pertanyaan saja melainkan selalu berisi banyak item (*multiple item measure*) maupun lingkungan dalam konteks yang dimaksudkan penggunaan.

Prosedur penskalaan dengan metode Likert didasari oleh dua asumsi, yaitu:

- a. Setiap pernyataan sikap yang telah ditulis dapat disepakati sebagai termasuk pernyataan yang *favorable* atau pernyataan yang tidak *favorable*,
- b. Untuk pernyataan positif, jawaban yang diberikan oleh individu yang memiliki sikap positif harus diberi bobot atau nilai yang lebih tinggi dari jawaban yang diberikan oleh responden yang mempunyai sikap negatif. Demikian sebaliknya untuk pernyataan negatif, jawaban yang diberikan oleh individu yang memiliki sikap negatif harus diberi bobot atau nilai yang lebih tinggi dari jawaban yang diberikan oleh responden yang mempunyai sikap positif.

Biasanya dalam skala Likert terbagi dalam lima kategori yang digunakan, tetapi banyak pakar psikometri yang menggunakan tujuh sampai sembilan kategori. Lima kategori tersebut adalah:

Tabel 1. Lima kategori skala likert

Pertanyaan Positif	Pertanyaan Negatif
(1)	(2)
1. Sangat tidak setuju	1. Sangat setuju
2. Setuju	2. Setuju
3. Ragu-ragu	3. Ragu-ragu
4. Setuju	4. Tidak setuju
5. Sangat setuju	5. Sangat tidak setuju





Adakalanya skala yang dipilih tidak sempurna dan memiliki potensi terjadinya kesalahan dalam pengukuran skala sikap. Oleh karena itu, perlu dilakukan uji validitas dan reliabilitasnya. Uji validitas digunakan untuk memastikan kemampuan skala (yang dipilih peneliti) dalam mengukur konsep yang bersifat abstrak. Sedangkan uji reliabilitas digunakan untuk memastikan bahwa pengukuran mengindikasikan keakuratan, stabilitas dan konsistensi dengan konsep yang diukur. Metode pengujian validitas dan reliabilitas akan dijelaskan secara lebih detail di bab metodologi.

Metode Statistik

Walpole (1993) menyebutkan metode statistik adalah suatu cara atau ilmu yang merupakan kumpulan dari aturan-aturan tentang pengumpulan, pengolahan, penaksiran, dan penarikan kesimpulan dari data statistik yang berupa angka-angka kuantitatif dan/atau kualitatif.

Statistik terbagi menjadi dua, yakni statistik deskriptif dan inferensia. Statistik deskriptif hanya menguraikan mengenai sesuatu keadaan atau masalah, sedangkan inferensia juga dibagi dua, yaitu teori perkiraan (*estimation theory*) dan hipotesis pengujian (*testing hypothesis*) yang merupakan pengambilan keputusan karakteristik populasi berdasarkan sampel. Dalam publikasi ini hanya digunakan statistik deskriptif untuk menggambarkan penerimaan dan evaluasi sistem CAPI pada PKL 54 STIS.



1. Statistik Deskriptif

Statistika Deskriptif adalah suatu ilmu yang merupakan kumpulan dari aturan-aturan tentang pengumpulan, pengolahan, penaksiran, dan penarikan kesimpulan dari data statistik untuk menguraikan suatu masalah.

Statistik deskriptif hanya memberikan informasi mengenai data yang ada dan sama sekali tidak menarik inferensia atau kesimpulan apapun tentang gugus induknya yang lebih besar. Contoh statistik deskriptif yang sering muncul adalah tabel, diagram, grafik, dan besaran-besaran lain di majalah dan koran-koran. Dengan statistik deskriptif, kumpulan data yang diperoleh akan tersaji dengan ringkas dan rapi serta dapat memberikan informasi inti dari kumpulan data yang ada. Informasi yang dapat diperoleh dari statistika deskriptif ini antara lain ukuran pemusatan data, ukuran penyebaran data, serta kecenderungan suatu gugus data.

Beberapa hal yang dapat dilakukan adalah penyajian data melalui tabel, grafik, diagram lingkaran, *pictogram*, perhitungan modus, median, mean, desil, persentil, perhitungan penyebaran data melalui perhitungan rata-rata dan standar deviasi, perhitungan persentase. Dapat juga dilakukan analisis korelasi antar variabel, analisis regresi atau membandingkan dua nilai rata-rata sampel/populasi.

Walpole (1993) menyebutkan ukuran pemusatan data adalah sembarang ukuran yang menunjukkan pusat segugus data, yang telah diurutkan dari yang terkecil sampai yang terbesar atau sebaliknya dari yang terbesar sampai yang terkecil. Penyelidikan segugus data kuantitatif dapat dibantu dengan menggunakan ukuran-ukuran numerik yang menjelaskan





ciri-ciri data yang penting. Ukuran penyebaran data diperlukan untuk mengetahui seberapa jauh suatu pengamatan menyebar dari rata-ratanya. Ukuran pemusatan data yang paling banyak digunakan adalah nilai tengah, median, dan modus. Masing-masing dari ukuran data tersebut memiliki kekurangan.

Beberapa ukuran pemusatan dan penyebaran data antara lain:

a. Mean

Mean (nilai rata-rata dari suatu gugus data) merupakan suatu ukuran pusat data bila data itu diurutkan dari yang terkecil sampai yang terbesar atau sebaliknya. Mean diperoleh dengan cara menjumlahkan semua nilai yang ada pada suatu gugus data yang kemudian dibagi dengan banyaknya nilai tersebut. Mean akan sangat dipengaruhi nilai pencilan.

b. Modus

Modus merupakan suatu ukuran pusat data dengan cara diperoleh melalui mencari nilai yang banyak keluar. Modus hanya dapat diterapkan dalam data dengan ukuran yang besar.

c. Standar deviasi

Standar deviasi adalah akar dari tengah kuadrat simpangan dan nilai tengah atau akar simpangan rata-rata kuadrat. Menurut Sudjana (1991) Standar deviasi merupakan ukuran penyebaran yang paling banyak digunakan. Semua gugus data dipertimbangkan sehingga lebih stabil dibandingkan dengan ukuran lainnya.

Secara garis besar ada dua cara penyajian data, yaitu dengan daftar atau tabel dan diagram atau grafik. Tabel atau daftar merupakan kumpulan



angka yang disusun menurut kategori atau karakteristik data sehingga memudahkan untuk analisis data. Berdasarkan Buku Metode Statistik karya Rasyidin Rasyad (2003) terdapat beberapa jenis grafik, grafik yang digunakan pada laporan ini adalah :

- a. Grafik batang/balok (*bar chart*) pada dasarnya sama fungsinya dengan grafik garis yaitu untuk menggambarkan data berkala.
- b. Grafik lingkaran (*pie chart*) lebih cocok untuk menyajikan data *cross section*, di mana data tersebut dapat dijadikan bentuk persentase.

2.2 Penelitian Terkait

Penelitian Praktik Kerja Lapangan Tahun Akademik 2015/2016 berbeda dengan tahun-tahun sebelumnya yang menggunakan CAPI bersifat statis, PKL 55 menggunakan CAPI yang bersifat dinamis dengan mengambil total sampel 100 persen untuk wisatawan dan 50 persen untuk unit usaha dan rumah tangga. CAPI PKL 55 dikatakan dinamis karena CAPI ini dapat menampung banyak kuesioner sekaligus. Aplikasi CAPI sebelumnya hanya bisa menampung satu jenis kuesioner sehingga jika terjadi perubahan kuesioner mereka juga melakukan perubahan pada aplikasi CAPI.

Listing dan penarikan sampel juga dapat dilakukan dalam aplikasi CAPI PKL 55. Listing kini dapat dilaksanakan langsung di lapangan dengan memasukkan identitas dan karakteristik unit observasi dalam aplikasi CAPI. Aplikasi CAPI juga dapat langsung menarik sampel sesuai kaidah yang berlaku. CAPI PKL 55 juga dapat menghasilkan *real-time* analisis atau dapat dipantau





perkembangannya secara langsung 24 jam tanpa harus menunggu mengumpulkan seluruh data untuk divalidasi.



Halaman ini sengaja dikosongkan



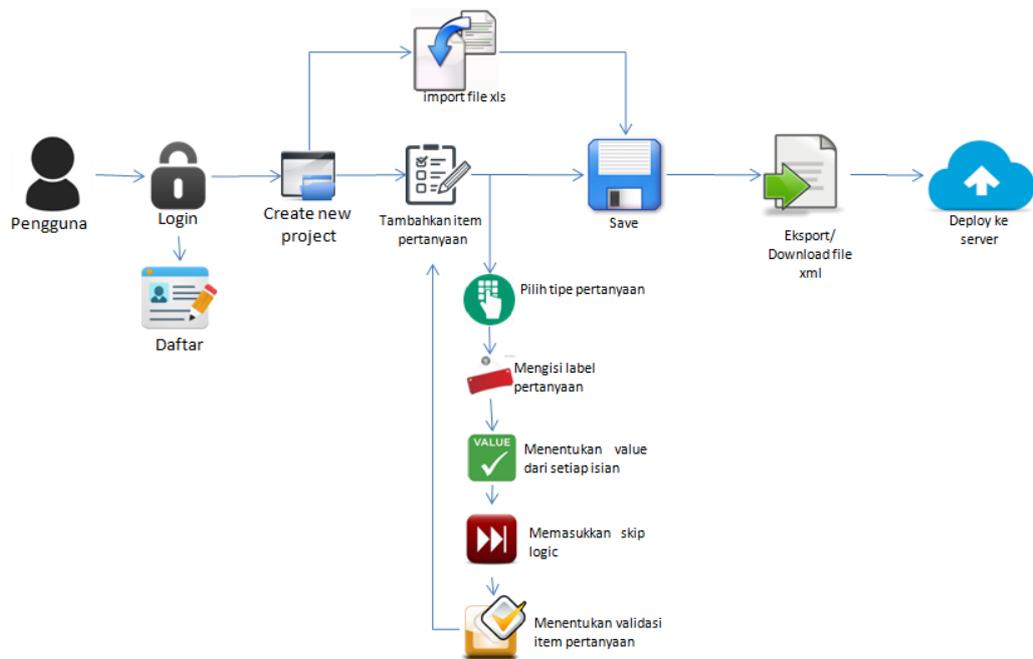


BAB III

PEMBANGUNAN SISTEM

3.1. Build

Alur Kerja



Gambar 1. Alur kerja subsistem *build* kuesioner aplikasi CAPI

Alur kerja subsistem *build* kuesioner adalah sebagai berikut:

1. Pengguna melakukan *login* untuk masuk ke sistem *build* kuesioner yang dalam hal ini menggunakan *kobotoolbox*. Untuk bisa *login* pengguna harus sudah terdaftar. Oleh karena itu, untuk pengguna yang belum mendaftar bisa melakukan proses *sign up* terlebih dahulu.
2. Setelah *login* berhasil, maka pengguna dapat memulai membuat kuesioner. Kuesioner dapat langsung di-*import* dari *file* .xls yang telah berisi berbagai



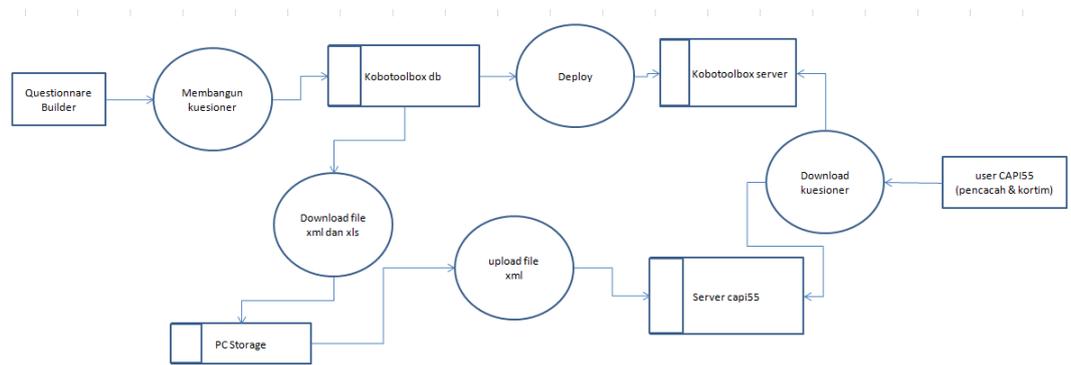


- item pertanyaan. Selanjutnya bisa disimpan dan menghasilkan *file .xml* yang kemudian di-*upload* ke server sebagai kuesioner yang telah siap digunakan.
3. Membuat kuesioner baru dengan tanpa meng-*import file .xls* bisa dilakukan dengan melakukan beberapa proses.
 - a. Memilih tipe pertanyaan, tipe pertanyaan pada subsistem *build* ini terdiri dari berbagai macam tipe. Beberapa contoh tipe yang digunakan adalah tipe *multiple choice*, *text*, *number*, *single choice*, penghitungan, dan sebagainya.
 - b. Mengisi label pertanyaan, label pertanyaan merupakan kolom yang menampilkan teks pertanyaan.
 - c. Menentukan *value* dari setiap isian, langkah ini dilakukan pada beberapa tipe item pertanyaan. Seperti *single choice* dan *multiple choice*, perlu dilakukan defenisi *value* pada setiap pilihan yang disediakan.
 - d. *Skip logic, field* ini digunakan untuk memasukan fungsi *skip*. Maksudnya item pertanyaan hanya akan muncul apabila fungsi yang diisi pada *skip logic* terpenuhi.
 - e. Menentukan validasi. Validasi digunakan untuk mendefenisikan kondisi yang tidak mungkin dari suatu item pertanyaan. Sehingga apabila pencacah melakukan input nilai yang salah atau tidak sesuai dengan validasi, pencacah tidak bisa melanjutkan pengisian ke item pertanyaan berikutnya hingga isian yang diinputkan benar-benar sesuai dengan validasi yang didefenisikan tersebut.
 4. Menyimpan proyek dan meng-*eksport*-nya ke dalam *.xml*.
 5. Melakukan *upload* kuesioner yang telah jadi (format *.xml*) ke server CAPI 55.





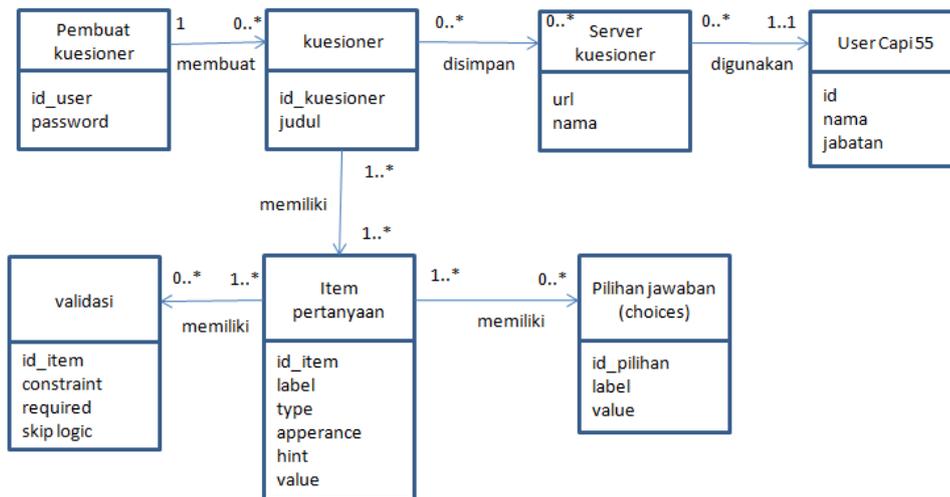
Data Flow Diagram



Gambar 2. *Data flow diagram* subsistem *build* CAPI 55

Gambar di atas menunjukkan arus data subsistem *build* pada aplikasi CAPI yang digunakan untuk pembuatan kuesioner yang digunakan pada PKL 55.

Entity Relationship Diagram



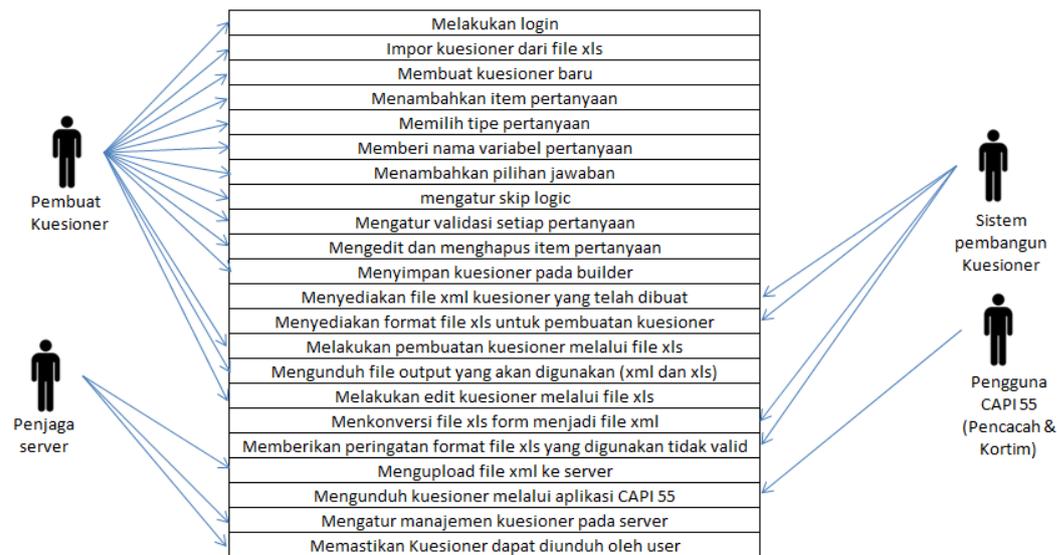
Gambar 3. *Entity relationship diagram* subsistem *build* aplikasi CAPI

Terdapat beberapa hubungan antar entitas pada diagram ERD di atas. Pembuat kuesioner dapat membuat satu atau lebih kuesioner. Dimana setiap



kuesioner memiliki minimal satu item pertanyaan. Setiap item pertanyaan dapat memiliki beberapa pilihan jawaban maupun tidak (jawaban berupa teks). Item pertanyaan juga bisa memiliki beberapa validasi. Setiap kuesioner yang sudah jadi, akan disimpan di server CAPI 55 untuk dapat digunakan oleh *user* CAPI 55 baik pencacah maupun kortim dengan cara *men-download-nya*.

Use Case



Gambar 4. Use case diagram pada subsistem *build* aplikasi CAPI 55

Terdapat empat aktor yang berperan dalam pembangunan kuesioner pada aplikasi CAPI 55. Pembuat kuesioner berperan dalam melakukan aktivitas yang berkaitan dengan pembuatan kuesioner baik membuat kuesioner baru maupun berdasarkan file xls-form yang sudah ada. Kegiatan yang paling penting dari pembuat kuesioner adalah memberi nama variabel setiap pertanyaan, menambahkan pilihan jawaban, menentukan *skip logic*, validasi kuesioner. Sistem pembangun server membantu proses konversi *file* .xls menjadi .xml serta





menyediakan *file* output xml yang dibutuhkan. Penjaga server meng-*upload file* .xml yang dihasilkan ke server serta melakukan manajemen kuesioner di server dan memastikan kuesioner dapat diunduh oleh *user*.

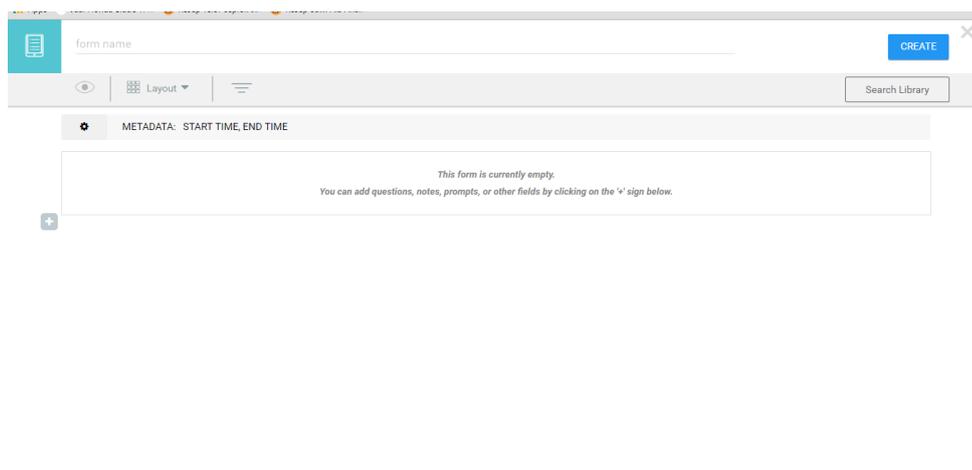
Implementasi Sistem

1. Proses *Build* Kuesioner

Proses *Build* kuesioner dapat dilakukan dengan dua cara, yaitu melalui sistem web seperti *kobotoolbox* dan melalui *.xls form*.

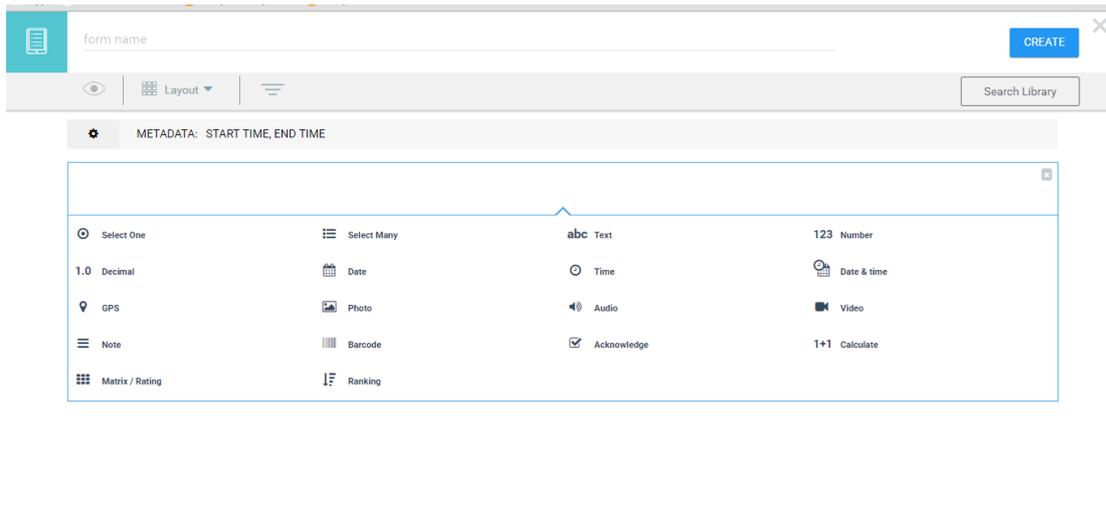
2. Pembangunan kuesioner melalui web *kobotoolbox* adalah sebagai berikut:

- a. Lakukan proses *login* pada web <http://www.kobotoolbox.org/>
- b. Setelah berhasil *login*, klik *create new form – NEW Form*
- c. Akan dimunculkan lembar kerja sebagai berikut:



Gambar 5. Halaman pertama *form* pembuatan kuesioner

- d. Klik *icon +* untuk menambahkan item pertanyaan
- e. klik “*add questions*” kemudian pilih tipe pertanyaan.



Gambar 6. Tampilan tipe pertanyaan pada pembuatan kuesioner CAPI

Berikut disajikan penjelasan mengenai tipe pertanyaan pada pembuatan kuesioner CAPI.

Tabel 2. Tipe pertanyaan dalam pembuatan kuesioner CAPI

<i>Select One</i>	:	Pertanyaan dengan beberapa pilihan jawaban. Pada pertanyaan ini hanya diperlukan satu jawaban.
<i>Select Many</i>	:	Pertanyaan dengan beberapa pilihan jawaban. Pada pertanyaan ini dapat dipilih beberapa jawaban.
<i>Text</i>	:	Disediakan kolom untuk menuliskan jawaban berupa teks
<i>Number</i>	:	Disediakan kolom untuk menuliskan jawaban berupa angka integer.
<i>Decimal</i>	:	Disediakan kolom untuk menuliskan jawaban berupa angka desimal.
<i>Calculate</i>	:	Tipe pertanyaan yang langsung diisi oleh sistem setelah <i>user</i> menginputkan nilai dari pertanyaan sebelumnya.





- f. Misalkan dipilih tipe pertanyaan *Select One*. Ketikkan label pertanyaan dan masukkan pilahan jawaban pada *option1*, *option2*, dan seterusnya. Jangan lupa menentukan *value* (nilai) pada setiap opsi. *Value* ini akan menjadi input pada isian ketika pilihan tersebut dipilih.
- g. Selanjutnya lakukan pengaturan terhadap nama variabel, validasi, dan *skip logic* dengan mengklik *icon* pengaturan.

The screenshot shows the 'Settings' panel for a question. The left sidebar contains 'Settings', 'Question Options', 'Skip Logic', and 'Validation Criteria'. The main area contains the following fields:

- Data Column Name:
- Question Hint:
- Mandatory Response: Yes
- Default Response:
- Appearance (Advanced):

Gambar 7. Pengaturan pertanyaan dalam pembuatan kuesioner CAPI

The screenshot shows the 'Validation Criteria' panel in the settings interface. The left sidebar is expanded to 'Validation Criteria'. The main area contains the following fields:

- Validation Code:
- Error Message:

Gambar 8. Pengaturan validasi dalam pembuatan kuesioner CAPI



Berikut disajikan mengenai pengaturan validasi:

Tabel 3. Tabel pengaturan validasi dalam pembuatan kuesioner CAPI

<i>Data Column Name</i>	:	Masukkan nama kolom atau nama variabel pertanyaan. Misal: b1r1
<i>Question Hint</i>	:	Isikan petunjuk pertanyaan. Misal: isikan 1 bila ya, 2 bila tidak
<i>Default Response</i>	:	Isian yang udah ada sebelum <i>user</i> melakukan input
<i>Apperance</i>	:	Pilihan tampilan pertanyaan. <i>Appearance</i> yang digunakan pada kuesioner PKL 55 adalah minimal, <i>likert, compact</i> .
<i>Skip logic</i>	:	Mengisikan kondisi <i>skip</i> , apabila kondisi yang ditentukan
<i>Validation Code</i>	:	<i>Field</i> untuk memasukkan kode validasi
<i>Error Message</i>	:	<i>Field</i> untuk memasukkan pesan <i>error</i> bila input tidak valid

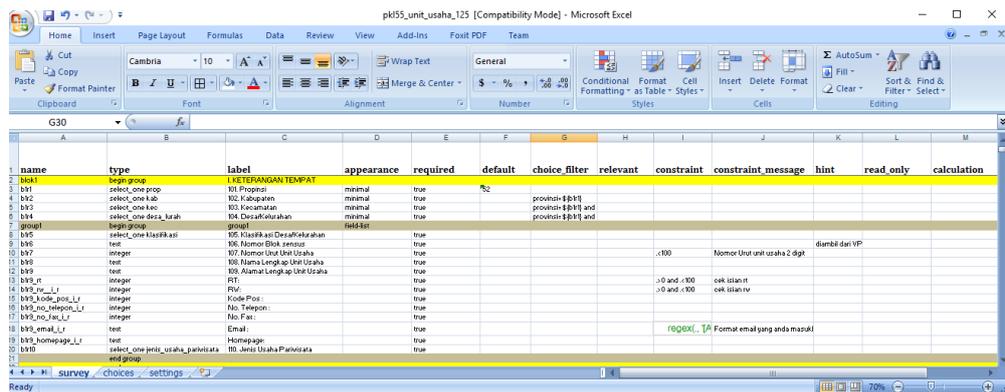
- h. Ulang langkah di atas untuk item pertanyaan berikutnya.
- i. Bila selesai, klik tombol *CREATE* pada pojok kanan atas.
- j. Lalu simpan kuesioner dengan tombol *SAVE*. Catatan: jangan lupa memberi nama kuesioner terlebih dahulu pada *field* 'form name'.
- k. Lalu keluar dari jendela kerja.
- l. Lakukan *download* file .xml yang dihasilkan pada *form* kuesioner.
- m. Form kuesioner siap di-*upload* ke server.



Dari segi kepraktisan, cara di atas cukup mudah dan memiliki *user interface* yang baik. Namun, pada kuesioner yang sifatnya besar atau memiliki pertanyaan yang banyak, cara tersebut sangat tidak efisien. Maka dari itu, pada PKL 55 *form* kuesioner dilakukan melalui *.xls form* yang di-*edit* melalui program Microsoft Excel.

Adapun pembangunan kuesioner melalui *.xls form* adalah sebagai berikut:

- a. Buka aplikasi Microsoft Excel.
- b. Sediakan tiga *sheet* dengan nama '*survey*', '*choices*', dan '*settings*'
- c. *Sheet survey* berfungsi untuk mendefinisikan tipe pertanyaan, label, besera validasinya. *Sheet choices* berfungsi untuk memasukkan pilihan pada tipe pertanyaan *single choice* dan *multiple choices*.
- d. Pada *sheet survey*, sediakan beberapa kolom seperti berikut:



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	name	type	label	appearance	required	default	choice_filter	relevant	constraint	constraint_message	hint	read_only	calculation
2	N041	begin_group	LAKETERANGAN TEMPAT										
3	N041	select_one_prop	101. Propinsi	minimal	true	%2							
4	N042	select_one_kab	102. Kabupaten	minimal	true		provinsi=101						
5	N043	select_one_kec	103. Kecamatan	minimal	true		provinsi=101 and kabupaten=102						
6	N044	select_one_desa_kelurahan	104. Desa/kelurahan	minimal	true		provinsi=101 and kabupaten=102 and kecamatan=103						
7		begin_group	RHSKBT										
8	N045	select_one_klasifikasi	105. Klasifikasi Desa/kelurahan		true								
9	N046	text	106. Nomor Blok sensus		true								
10	N047	integer	107. Nomor Unit Usaha		true				>.000	Nomor Unit usaha 2 digit	dibuat dari VP		
11	N048	text	108. Nama Lengkap Unit Usaha		true								
12	N049	text	109. Alamat Lengkap Unit Usaha		true								
13	N049_1	integer	109.1		true								
14	N049_1_1	integer	109.1.1		true								
15	N049_1_1_1	integer	109.1.1.1		true				>0 and <.100	cek listan rv			
16	N049_1_1_1_1	integer	109.1.1.1.1		true				>0 and <.100	cek listan rv			
17	N049_1_1_1_1_1	integer	109.1.1.1.1.1		true								
18	N049_email	text	Email		true								
19	N049_homepage	text	Homepage		true				regex: [^	Format email yang anda masuki			
20	N049	select_one jenis_usaha_pariwisata	100. Jenis Usaha Pariwisata		true								
21		end_group											

Gambar 9. Tampilan *worksheet survey* pada *.xls form*



Berikut disajikan keterangan untuk masing masing kolom:

Tabel 4. Tabel penjelasan *worksheet survey* pada *.xls form*

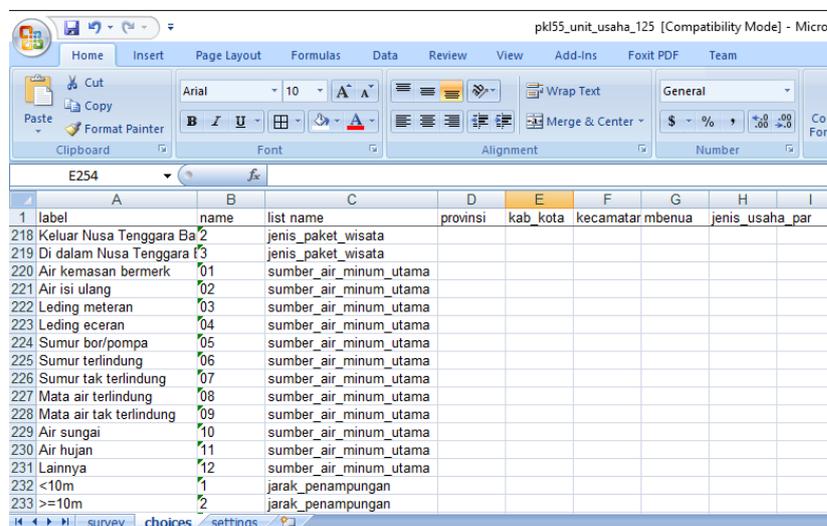
<i>Name</i>	:	Mendefenisikan nama variabel. Contoh nama: b1r1. Note: pada nama variabel ada yang ditambahkan ‘_i_r’ contoh: b1r9_r2_i_r. Tujuannya untuk melonggarkan <i>validasi required</i> dan <i>constraint</i> agar tetap bisa dilewati apabila input data di lapangan tidak sesuai dengan validasi yang ditetapkan.
<i>Type</i>	:	Mendefinisikan tipe pertanyaan.
<i>Label</i>	:	Pertanyaan yang muncul di aplikasi android
<i>Apperance</i>	:	Untuk pengaturan tampilan pertanyaan. Bila kosong maka pengaturannya <i>default</i>
<i>Required</i>	:	Validasi untuk menentukan bahwa isian dari pertanyaan tersebut boleh kosong atau tidak.
<i>Default</i>	:	Nilai awal yang ditetapkan sebelum <i>user</i> melakukan input data
<i>Choice_filter</i>	:	Menyaring pilihan (<i>choices</i>) yang ditampilkan pada aplikasi android bergantung pada pilihan sebelumnya. Contoh: pertanyaan b1r1 : provinsi DKI Jakarta, b1r2: kabupaten, pilihan yang dimunculkan adalah kabupaten yang berada di Provinsi DKI Jakarta.
<i>Relevant</i>	:	Tempat untuk mendefenisikan <i>skip logic</i> . Pertanyaan tersebut hanya akan muncul apabila kondisinya sesuai





	:	dengan apa yang didefenisikan di kolom ini.
<i>Constraint</i>	:	Validasi yang berfungsi membatasi nilai input.
<i>Constraint_message</i>	:	Pesan yang akan dimunculkan apabila input tidak sesuai dengan batasan yang telah ditetapkan.
<i>Hint</i>	:	Tempat untuk melampirkan petunjuk atas pertanyaan yang ditampilkan.
<i>Read_only</i>	:	Mendefinisikan bahwa pertanyaan tersebut hanya bisa dilihat (apabila <i>true</i>). <i>Read_only</i> biasanya digunakan pada tipe pertanyaan yang merupakan hasil kalkulasi/ penghitungan agar tidak bisa diubah oleh <i>user</i> .
<i>Calculation</i>	:	Memasukkan input secara otomatis berdasarkan input dari pertanyaan sebelumnya.

- e. Pada *sheet* ‘*choices*’ dimasukkan berbagai pilihan jawaban yang telah didefenisikan pada *sheet* ‘*survey*’ beserta dengan nilai inputnya.



Gambar 10. Tampilan *sheet* “*choices*” pada *.xls form*



- f. Pada *sheet settings* berguna untuk memberikan identitas pada kuesioner berupa *form id* dan judul kuesioner.

Fungsi-fungsi validasi pada Kuesioner CAPI

Tabel 5. Fungsi validasi pada kuesioner CAPI

Fungsi	Deskripsi	Contoh
(1)	(2)	(3)
And	Menggabungkan dua <i>statement</i>	$.<50$ and $.>30$ artinya 'pertanyaan ini' memiliki input kecil dari 50 dan besar dari 30
Or	memberikan opsi salah satu input yang valid	$.>6$ or $.<3$
.	memberikan rujukan pada dirinya sendiri.	
$\${b1r1}$	merujuk pada variabel <i>b1r1</i>	
Constraint	memberikan batasan input	$\${b1r1}<30$, artinya pertanyaan <i>b1r1</i> tidak bisa dilewati apabila inputnya lebih dari 30
Relevant	Pertanyaan akan	$\${b1r3}=2$. Pertanyaan hanya





	<i>skip</i> bila tidak memenuhi pernyataan yang didefenisikan di dalamnya.	akan muncul apabila <i>b1r3</i> bernilai 2
operator +,-,/,*	dapat digunakan pada <i>field calculation</i>	
selected($\{b1r1\}$,2)	mengatakan bahwa 'apabila pada pertanyaan <i>b1r1</i> bernilai 2' . fungsi ini dapat diletakkan pada <i>field constraint</i> maupun <i>relevant</i> sebagai fungsi validasi.	
if(statement,'true','false')	fungsi logika apabila <i>statement</i> sesuai maka akan bernilai <i>true</i> , bila tidak akan bernilai <i>false</i> .	if($\{b1r1\}=2$,'LOW','HIGH')



$<, >, =, !=, <=, >=$	lebih kecil, lebih besar, sama dengan, tidak sama dengan, kecil sama, besar sama.	
<code>not(selected(variabel,value))</code>	invers dari <i>selected</i> (variabel, <i>value</i>)	

3.2 Aggregate

ODK *Aggregate* menyediakan server yang siap di-*deploy* beserta data *repository*-nya untuk:

1. Menyediakan kuesioner kosong untuk ODK *Collect*
2. Menerima kuesioner Final
3. Memvisualisasikan data
4. *Export* dan publikasi data.

ODK *Aggregate* dapat di-*deploy* di Google App Engine ataupun Java Server lainnya, seperti tomcat server menggunakan MySQL atau PostgreSQL.

Cara Instalasi ODK *Aggregate* pada Tomcat

1. *Install* PostgreSQL atau MySQL





2. *Install* Tomcat 6.0. Jika menggunakan selain Tomcat 6.0 maka diperlukan konfigurasi tambahan pada context.cml yaitu pada line <context>, tambahkan variabel menjadi<context useHttpOnly=False>
3. Lalu *download* ODK *Aggregate* v1.N.N.di halaman *download* Open Data Kit.
4. Ikuti langkah-langkah yang ditunjukkan.
5. *Installer* ODK *Aggregate* sebenarnya tidak meng-*install* apapun, *installer* tersebut hanya membantu konfigurasi awal file .war dari *aggregate* agar konfigurasinya mudah. Pada akhirnya *installer* tersebut akan menghasilkan suatu file .war dan .db dimana keduanya harus di-*deploy* secara manual.

Catatan penting :

- a. Untuk membuat alamat *deploy aggregate* dapat diakses dari luar pc, gunakan IP *public PC*, jangan gunakan *local host*.
- b. Untuk mengganti alamat yang sudah di konfigurasi, unrar terlebih dahulu ODK *Aggregate-setting* yang ada di folder WEB-INF di direktori *aggregate*.
- c. Untuk membuat server dapat diakses dari Internet mintalah domain pada *server admin* yang meredirect pada IP *PC server*, lalu gunakan domain tersebut sebagai alamat *deploy aggregate*.

Penggunaan *Aggregate* ODK

1. Setelah instalasi di *server*, konfigurasi awal *Aggregate* ODK memungkinkan untuk melakukan *submissions* oleh pengguna yang tidak berkepentingan (*anonymousUser*) dari ODK *Collect* dan juga *anonymousUser* tersebut mengakses fungsi dan bentuk pengelolaan *Aggregate* ODK. Ketika URL ke





- server *Aggregate* ODK pertama kali dibuka, Anda akan disajikan dengan halaman aplikasi yang menunjukkan tab *Submissions* dan *Form Management*.
2. Data yang dikumpulkan di *Aggregate* ODK dapat dilihat, di-*ekspor*, dipetakan, dan dihapus.
 3. Gunakan tombol *Add New Form* pada *Form Management* untuk meng-*uploadform* baru ke *Aggregate* ODK.
 4. Lihat data yang disimpan di *Aggregate* ODK pada tab *Submissions*.
 5. Jika tab *Site Admin* tidak terlihat, klik *Log In* link di sudut kanan atas layar yang akan menyajikan halaman *Log onto Aggregate*. Pilih tombol *Sign in with Aggregate password* dan masukkan *super-user username* yang Anda tentukan ketika saat meng-*install*. Sandi awal untuk akun ini adalah *aggregate*. Ketika masuk dengan metode ini, jika Anda tidak memasukkan *password* dengan benar, Anda mungkin harus menutup semua jendela *browser* Anda dan menutup *browser* Anda kemudian dapat dicoba lagi.
 6. Jika Anda belum mengubah *password super-user*, server akan menampilkan "This server and its data are not secure! Please change the super-user's password!", pada bagian atas halaman web. Silakan klik subtab *Permissions* pada tab *Site Admin* untuk mengubah *password*.
 7. Jika nama *instance* dari server berubah, maka *password* untuk semua *username Aggregate* ODK akan dihapus (untuk mencegah pengguna yang lain) dan *password* dari *super-username* akan dikembalikan ke *aggregate* dan pesan (merah) di atas akan juga akan ditampilkan. Dalam hal ini, Anda harus *login*, mengubah *password super-user*, dan mengubah *password* untuk semua *username Aggregate* ODK Anda.





- Gunakan subtab *Permissions* pada tab *Site Admin* untuk membatasi siapa yang dapat mengunduh formulir atau mengirimkan data dari *Aggregate ODK*. Lakukan ini dengan membuat sebuah *username* dan *password ODK* dan memberikan hak sebagai *Data Collector*. *Username* dan *password* ini kemudian dapat dimasukkan ke halaman pengaturan *Aggregate ODK*. Ketika membatasi akses Anda juga harus menghapus hak *Data Collector* dari *anonymousUser*. Ingatlah untuk klik *Save Changes* untuk membuat perubahan ini berlaku. Sebaliknya, pemberian hak *Data Collector* kepada *anonymousUser* memungkinkan orang untuk mengirimkan data ke server *Aggregate ODK* Anda.

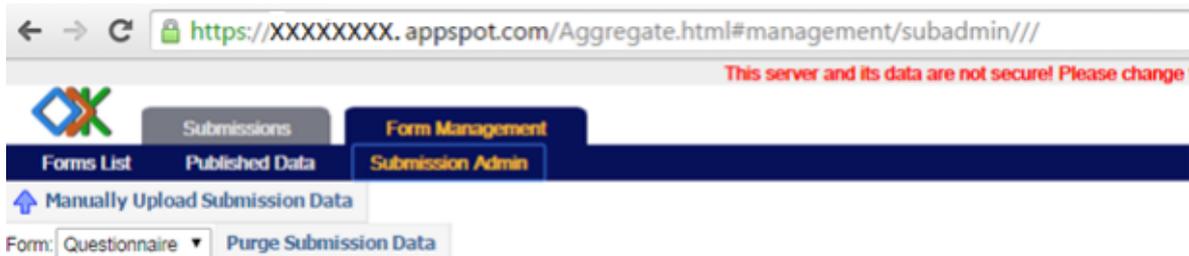
Gunakan sub-tab *Permissions* pada tab *Site Admin* untuk menentukan *username* tambahan untuk memberikan akses dari browser ke server. Untuk setiap pengguna yang Anda tambahkan, pilih apakah mereka memiliki akses ke data yang disimpan (hak *Data Viewer*), kemampuan untuk meng-*upload Form* dan *ekspor* atau mempublikasikan data (hak *Form Management*), atau kemampuan untuk mengelola akses situs dan pengguna (hak *Site Admin*). Ingatlah untuk klik *Save Changes* untuk membuat perubahan ini berlaku. Setiap nama pengguna yang Anda tambahkan, ingat untuk *Change Password* untuk menetapkan *password* secara default, *username* diciptakan dengan *password* yang tidak dapat digunakan. Pemberian setiap hak istimewa yang disebutkan sebelumnya ke *anonymousUser*, memungkinkan mereka mengakses dari *browser* untuk menjalankan fungsi-fungsi di *Aggregate ODK* tanpa terlebih dahulu *Log-in*.



9. Upload Form

Upload submission secara manual:

- a. Pertama tama klik *submission admin* di bawah *form management*



Gambar 11. Tampilan *Upload Submission Data*

- b. Lalu klik *Manually Upload Submission Data*

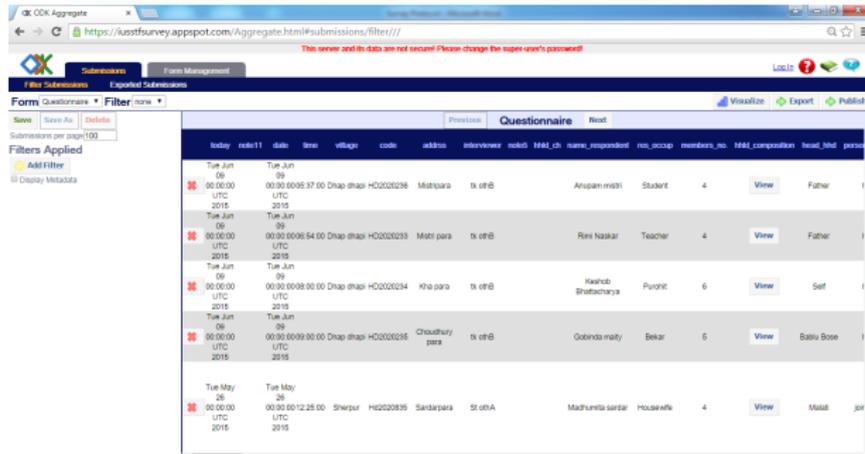


Gambar 12. Tampilan *Manually Upload Submission Data*



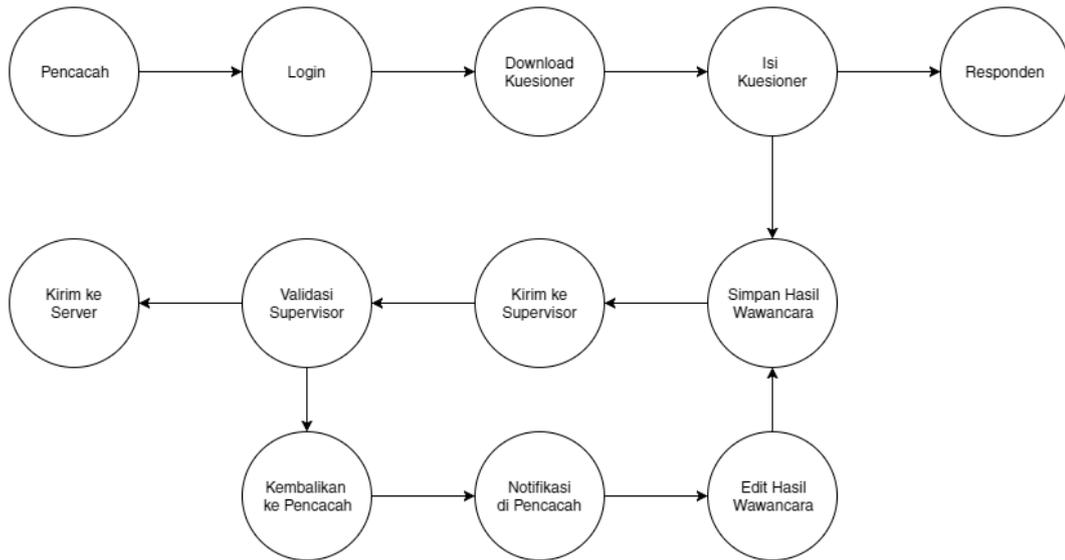


10. Pilih lokasi XML *Submission* yang ingin diunggah dan pilih *upload submission*.



Gambar 13. Tampilan apabila kuesioner sukses diunggah

3.3 Pengumpulan Data (Collect)



Gambar 14. Alur kerja aplikasi CAPI

1. Pencacah harus masuk ke aplikasi CAPI 55 sebagai pengguna yang telah terautentikasi.



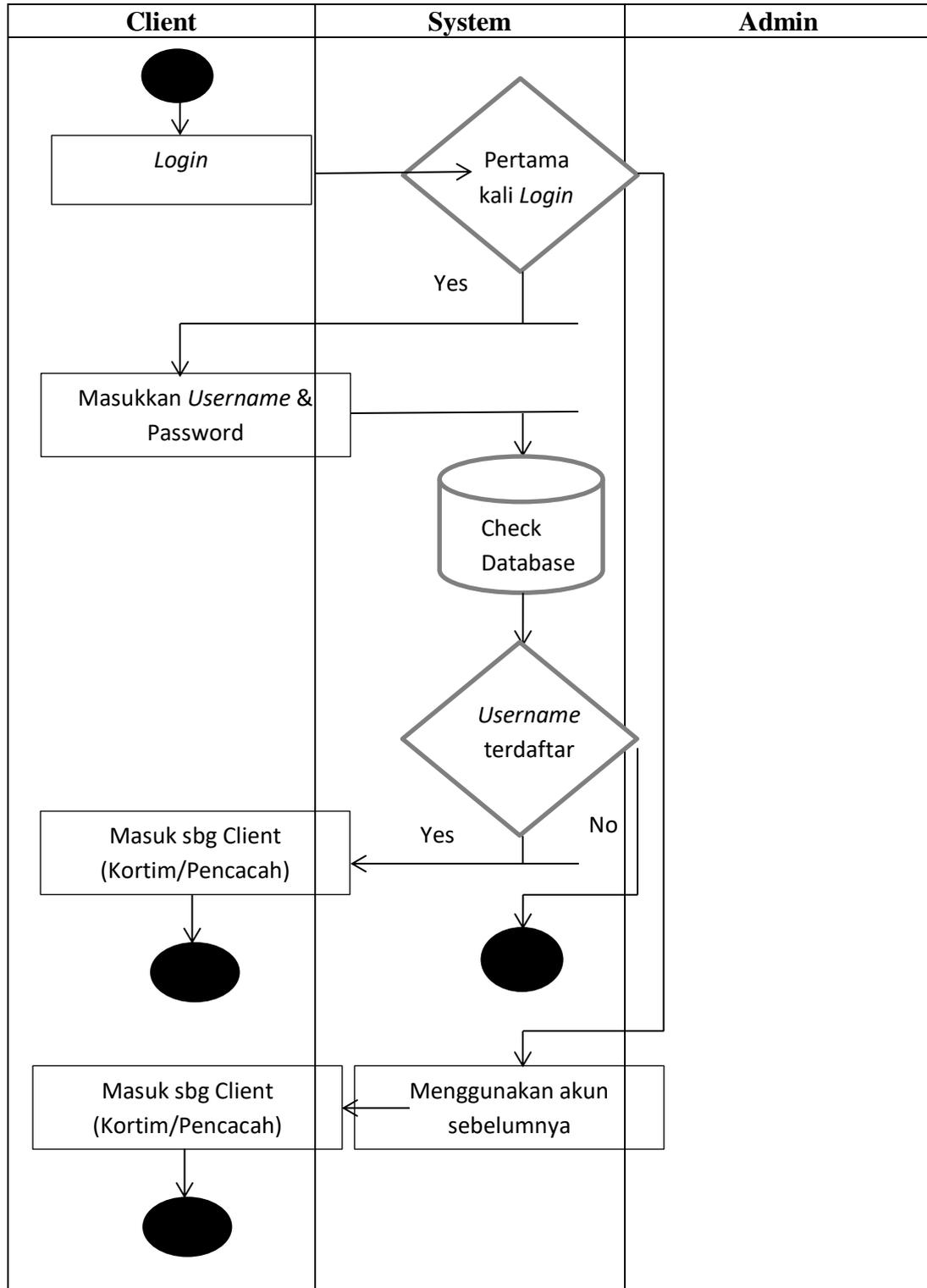
2. Saat pertama kali dipasang, CAPI 55 belum menyediakan kuesioner apapun maka dari itu pencacah harus mengunduh kuesioner dari server yang telah ditentukan.
3. Pilih kuesioner yang telah diunduh yang selanjutnya diisi ketika melakukan wawancara dengan responden.
4. Simpan hasil wawancara, kemudian hasil tersebut dikirimkan ke supervisor untuk dilakukan pengecekan (validasi).
5. Validasi dilakukan oleh supervisor terhadap isian pencacah ketika mewawancarai responden, apabila terdapat hal yang tidak sesuai maka hasil wawancara bisa dikembalikan ke pencacah.
6. Pencacah bisa melihat hasil wawancara yang dikembalikan dari supervisor dalam notifikasi dan kemudian dilakukan edit terhadap hasil wawancara dalam selanjutnya disimpan serta dikirimkan kembali ke supervisor.
7. Hasil wawancara yang telah sesuai selanjutnya akan dikirimkan ke server oleh supervisor.





Business Process

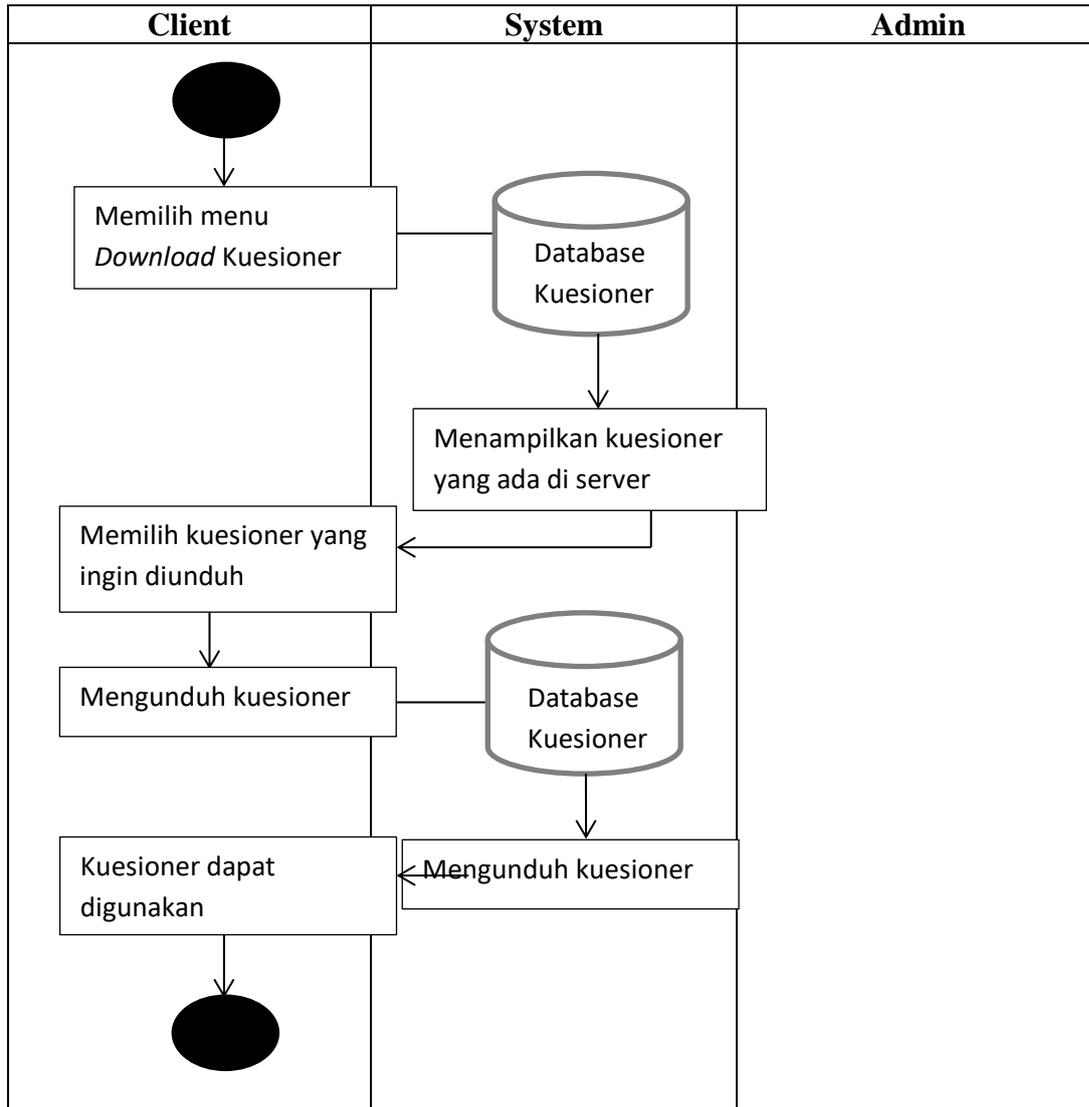
1. *Login*



Gambar 15. *Business Process* pada proses *Login*



2. *Download* Kuesioner

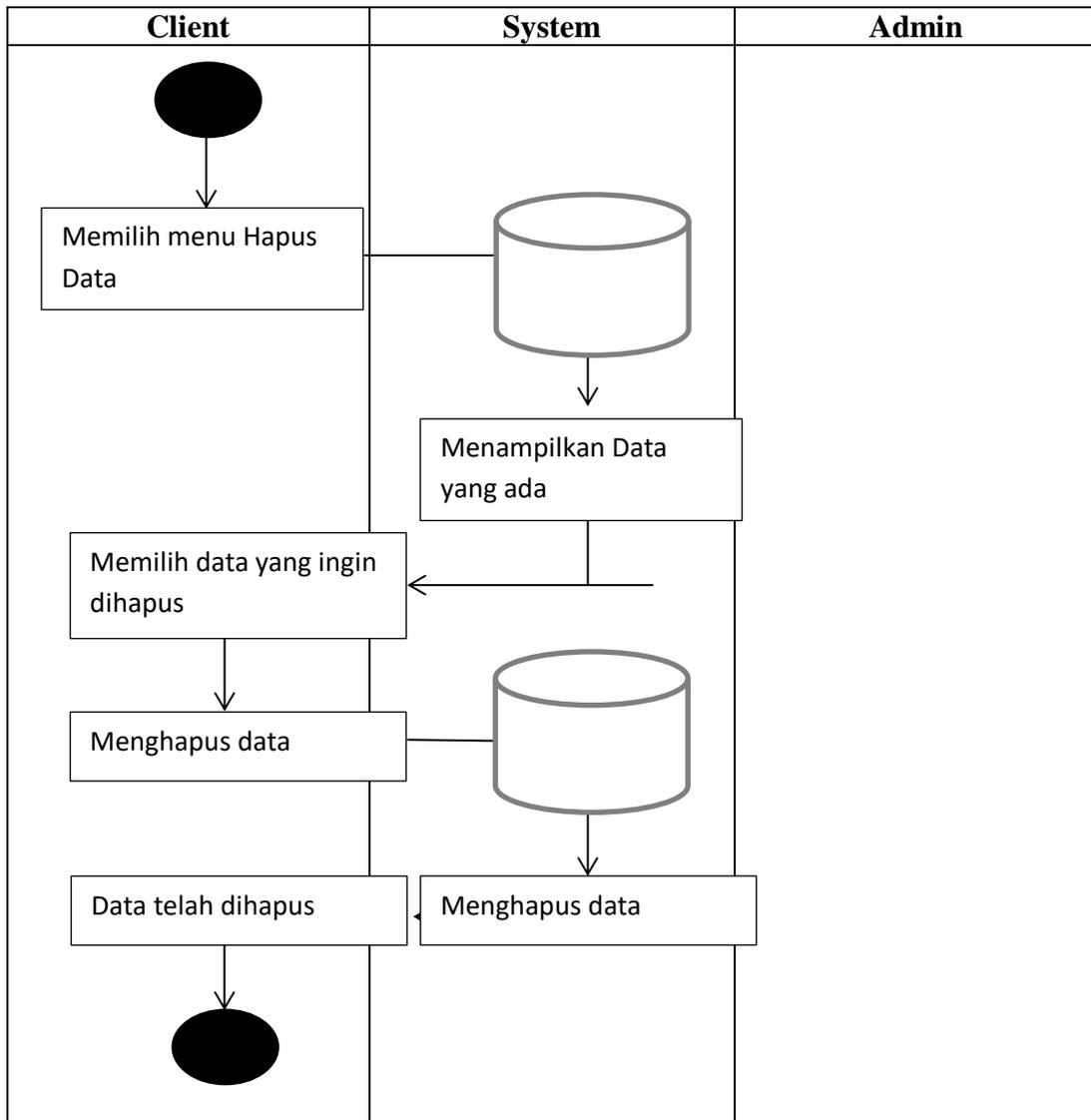


Gambar 16. *Bussines Process* pada proses *Download* Kuesioner





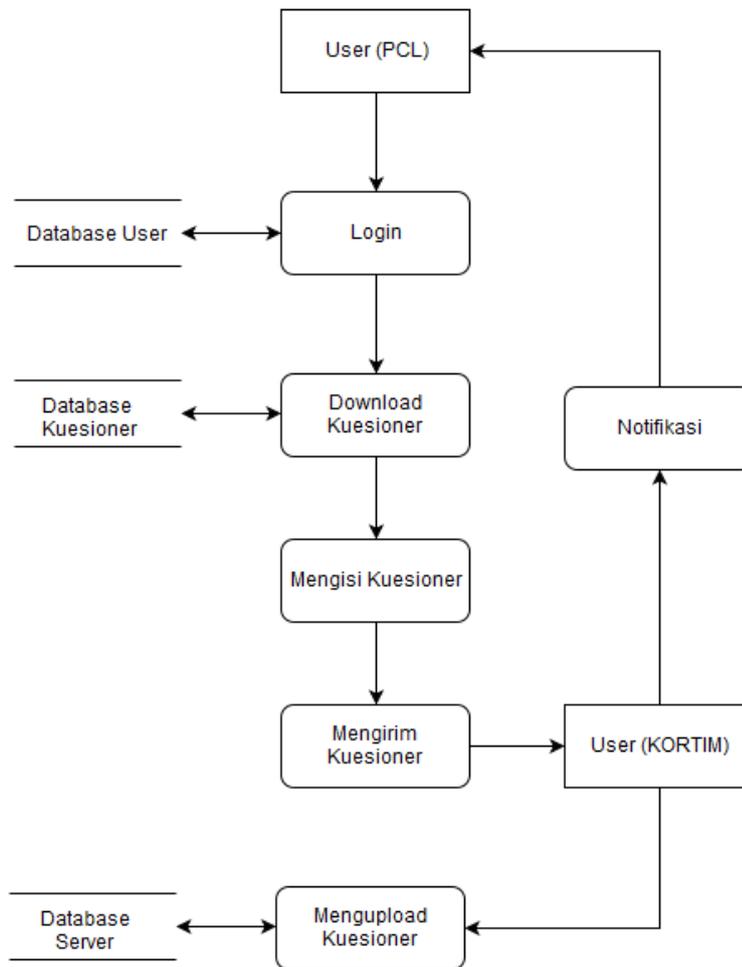
3. Hapus Data



Gambar 17. *Bussines Process* pada proses Menghapus Data Kuesioner



Data Flow Diagram (DFD)



Gambar 18. *Data Flow Diagram* pada aplikasi CAPI

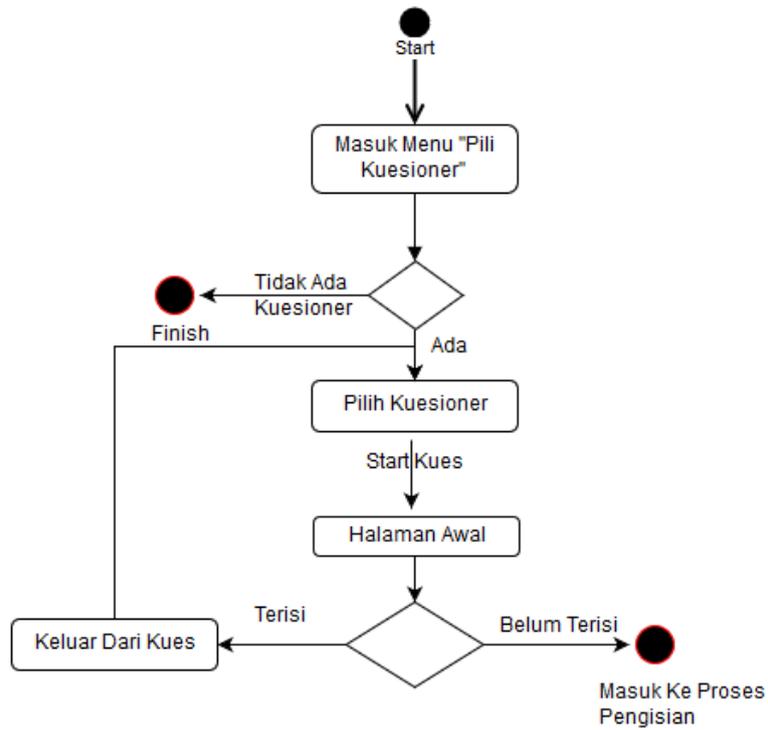
Diagram diatas menunjukkan aliran data dari sistem, dimulai dari *User* sebagai PCL melakukan proses *Login*. Kemudian, *user* (PCL) bisa mendownload kuesioner yang ada di databse kuesioner dan mengisi kuesioner. Setelah kuesioner selesai diisi, *user* (PCL) bisa mengirim kuesionernya ke *user* (Kortim) dan jika kuesionernya sudah lengkap dan benar, maka *user* (Kortim) memberikan notifikasi kepada *user* (PCL), lalu *user* (Kortim) meng-*upload* kuesionernya ke server.



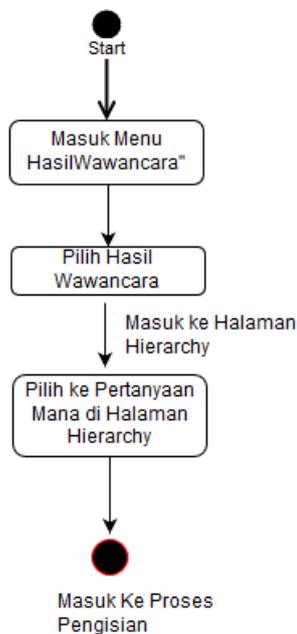


Activity Diagram

Isi Kuesioner dan Hasil Wawancara

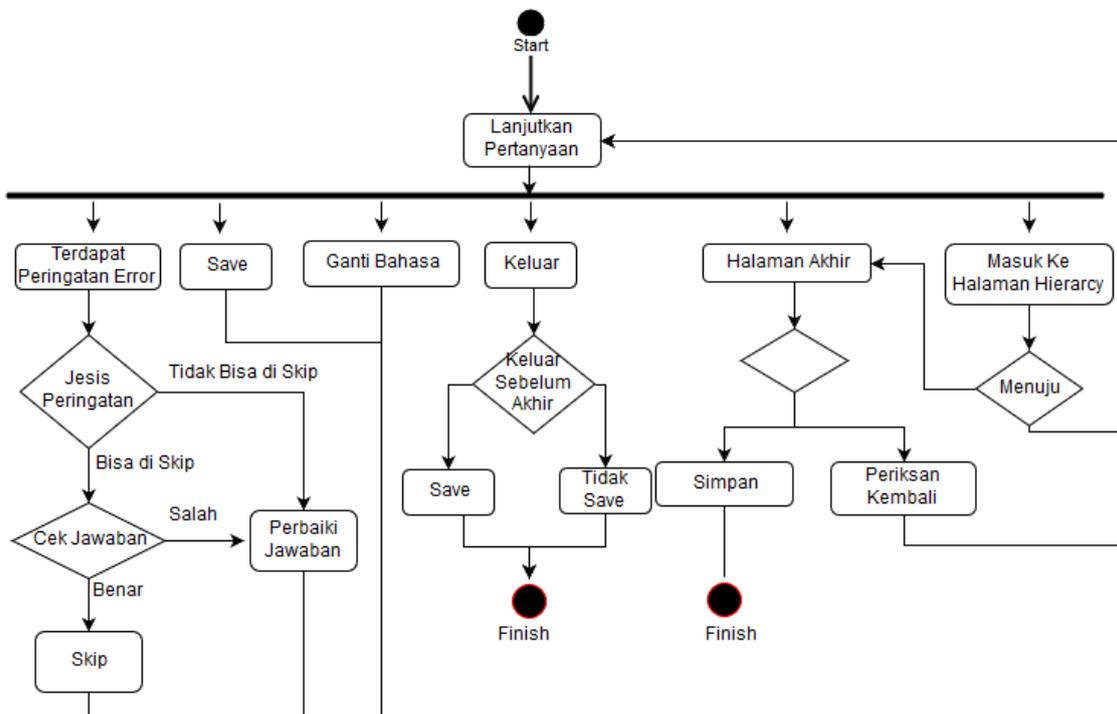


Gambar 19. *Activity Diagram* pada proses Pengisian Kuesioner



Gambar 20. *Activity Diagram* pada proses Penampilan Hasil atau Wawancara





Gambar 21. *Activity Diagram* pada proses Pengisian Wawancara

Pada gambar 19 adalah *Activity Diagram* yang terjadi *Activity Diagram* saat pertama kali masuk ke menu Isi Kuesioner. Saat masuk menu Pilih Kuesioner bada bagian ini, CAPI mengakses database untuk mendapatkan list Form yang tersedia menggunakan *class FormProvider*. Lalu jika tidak menemukan Kuesioner yang akan diisi maka *Stop*, jika ada maka diklik. Setelah masuk biasanya akan ada dua kemungkinan yang muncul, yang sudah terisi atau yang belum terisi, yang sudah teriisi ini sebenarnya adalah isian yang sebelumnya tidak disimpan oleh *user* saat menggunakan kuesioner tersebut. Pada saat *me-load* kuesioner digunakan *class FormLoadaerTask*, sedangkan saat sudah mulai menjalankan pertanyaan yang utama adalah *FormEntryActivity*, *FormController*, *FormHierarchyActivity*, *HierarchyListAdapter* dan *ItemstDbAdapter*.

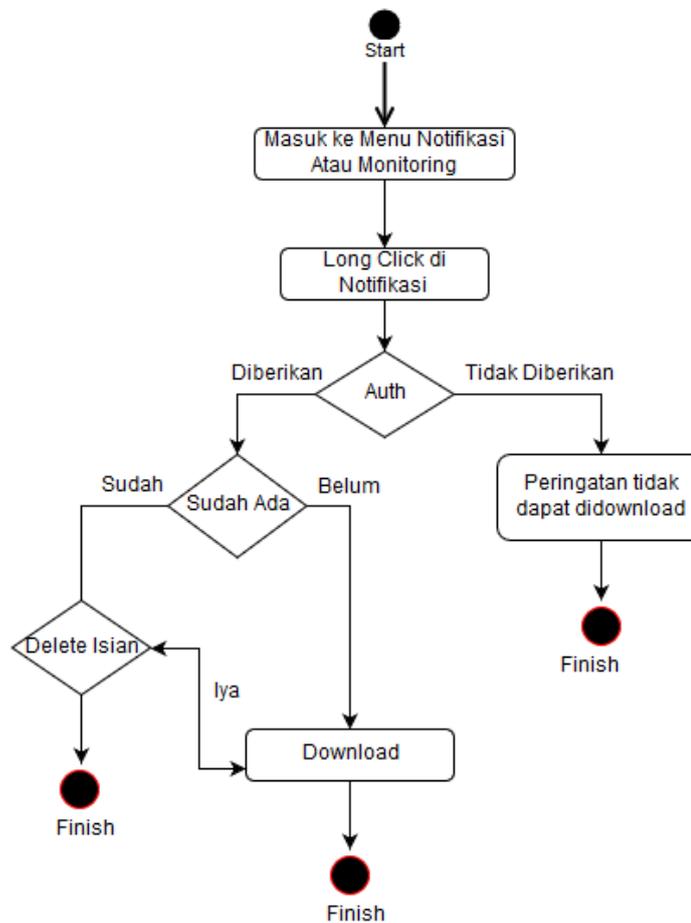




Proses pada gambar 19 hampir sama dengan gambar 20, tetapi tujuannya untuk membuka isian yang telah diisi dan disimpan sebelumnya. Namun, pada saat kita melakukan edit isian ini, tampilan pertama yang muncul bukan halaman awal, tetapi *Hierarchy view, class* yang digunakan sama seperti di atas.

Pada gambar 21 digambarkan proses pengisian kuesioner yang akan terjadi setelah tahap inisiasi (gambar 19), *class* yang digunakan juga hampir sama.

Download Isian



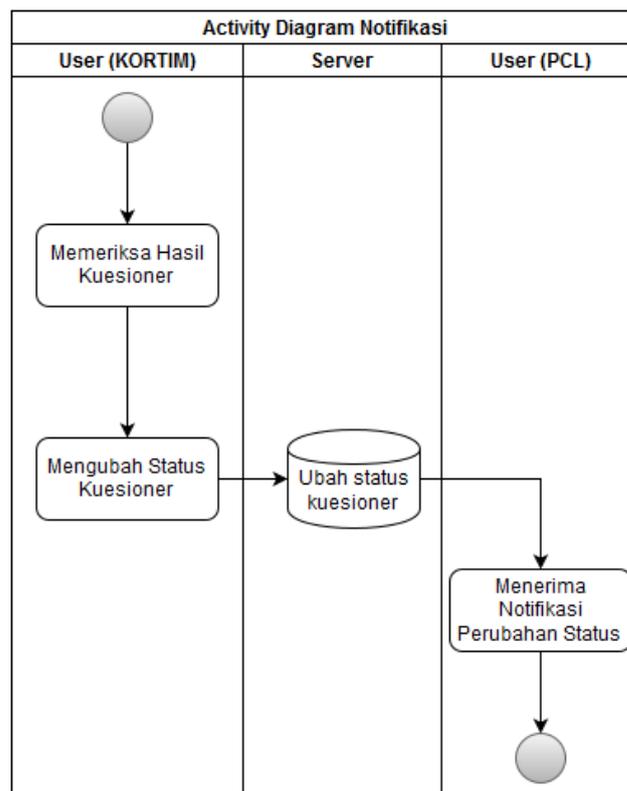
Gambar 22. *Activity Diagram* saat mengunduh isian kuesioner



Pada saat ingin mengunduh suatu dokumen kita harus masuk ke menu Notifikasi untuk PCL dan menu Monitoring untuk Kortim. Setelah masuk jika ada notifikasi mengenai isian, kita dapat mengunduhnya dengan *long click* di notifikasi isian yang diinginkan. Akan muncul apakah menurut akun *user*, apakah dia diberikan *authorisasi* mengenai *download* tersebut. Jika “iya”, maka akan diperiksa apakah isian sudah pernah diunduh atau tidak. Jika sudah akan muncul menu, jika “iya” baru terjadi *download*. Sedangkan jika sebelumnya belum terunduh maka akan terunduh langsung.

Class yang berpengaruh dalam proses ini adalah *class* yang masuk ke dalam proses notifikasi dan *class* yang ada di folder *DownloadBriefcase*.

Notifikasi



Gambar 23. *Activity Diagram* proses Notifikasi Perubahan Status

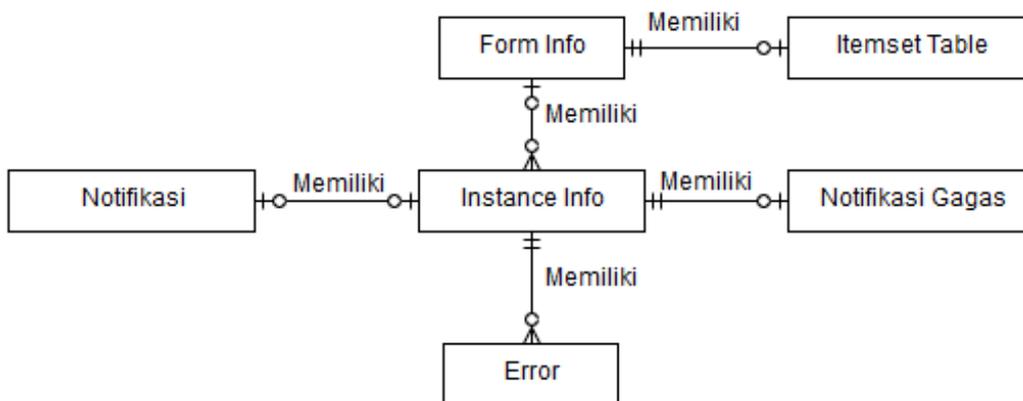




Diagram di atas menunjukkan cara kerja dari notifikasi. Awalnya, Kortim memeriksa hasil isian kuesioner dari PCL dan jika sudah benar, maka statusnya diubah menjadi *final clear* dan mengirimkan notifikasi kepada PCL bahwa kuesionernya sudah benar dan terunggah di server.

Class di java yang berpengaruh untuk memberikan notifikasi kepada user adalah *class* MyService.java pada folder *Notify*.

Entity Relationship Diagram



Gambar 24. *Entity Relationship Diagram* CAPI PKL 55

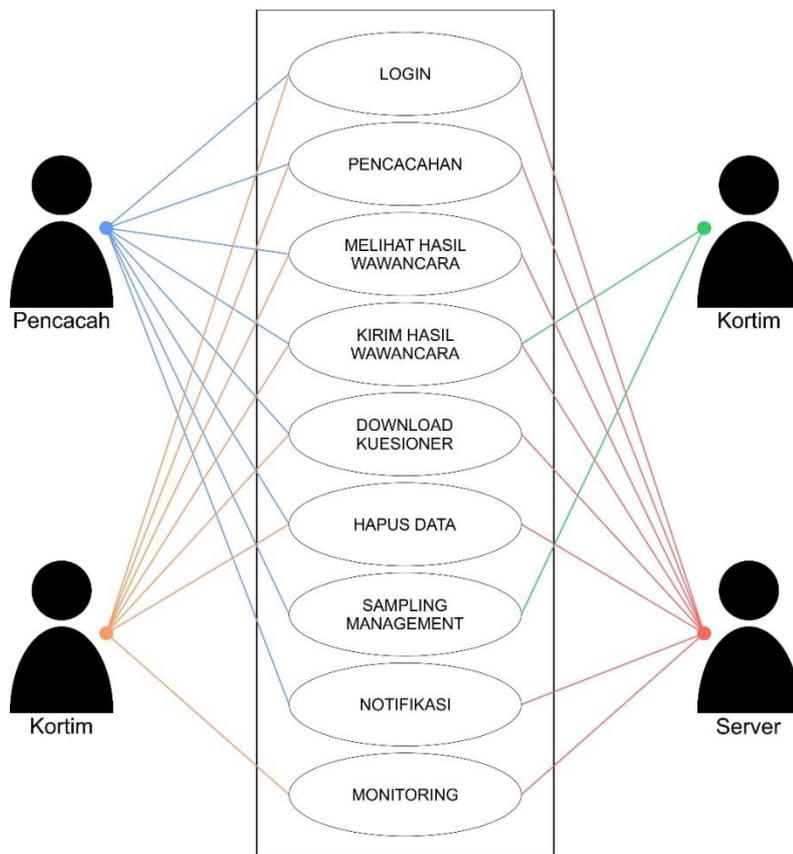
CAPI PKL 55 menyimpan isian (kuesioner yang telah diisi) sebagai .xml file, sehingga tidak ada database yang menyimpan isian ini secara langsung. Melainkan yang ada adalah database info yang menjadi kamus untuk kuesioner maupun isian.

Tabel *Form info* dan *Tabel Instance Info*, merupakan tabel penunjuk nama dan id dari form (kuesioner) maupun isian, juga merupakan penunjuk lokasi file .xml file baik untuk *form* dan *file* medianya maupun isian dan *file* medianya.



Tabel *itemset* menyediakan tabel untuk menyimpan nilai *itemset* .csv yang menjadi *external file*, yang dipakai di dalam ODK sendirinya. Tabel *Error* menyediakan informasi mengenai *Error* yang ditangkap waktu pengisian Kues, terutama yang dilangkahi. Tabel notifikasi bertugas menyimpan notifikasi yang diterima suatu akun dalam kurun waktu selama *login*, maupun tabel notifikasi gagal berisi informasi notifikasi yang gagal dikirim namun isiannya telah berhasil dikirim. *Class* yang bertanggung jawab untuk database sendiri terbagi-bagi, untuk *Instance info* terdapat di *class* InstanceProvider. Untuk *FormInfo* adalah *FormProvider*, sedangkan untuk yang lainnya di *ItemsetDbA*.

Use Case

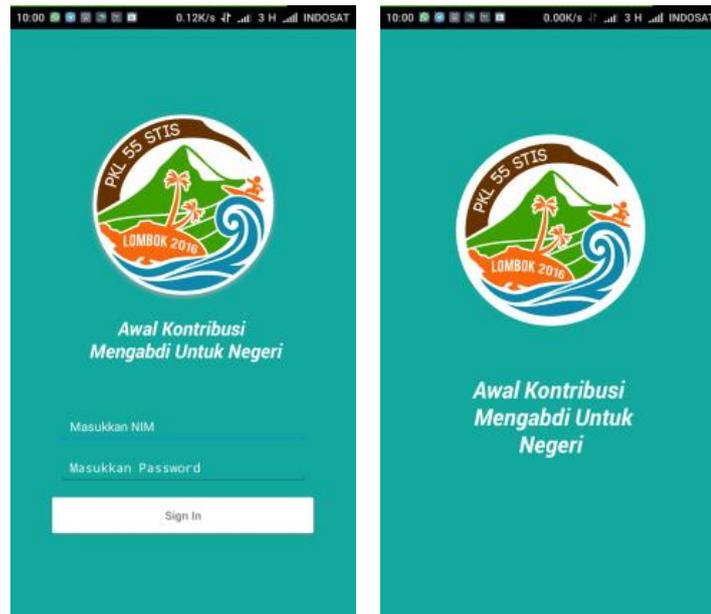


Gambar 25. Use Case dalam Aplikasi CAPI



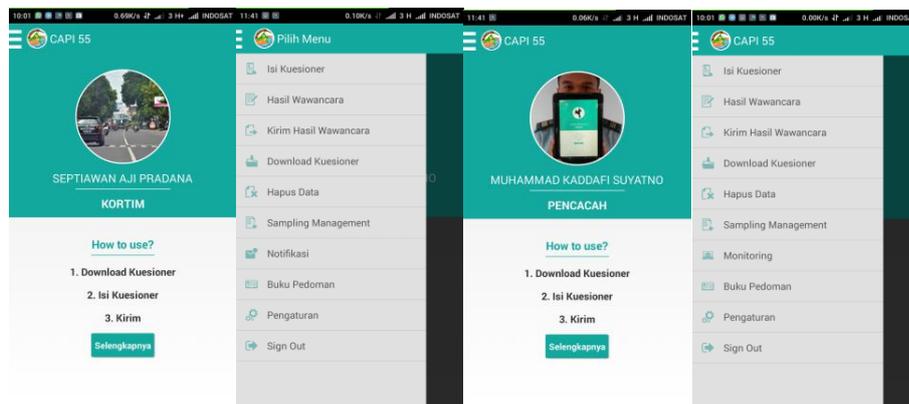


User Interface



Gambar 26. Tampilan depan aplikasi CAPI

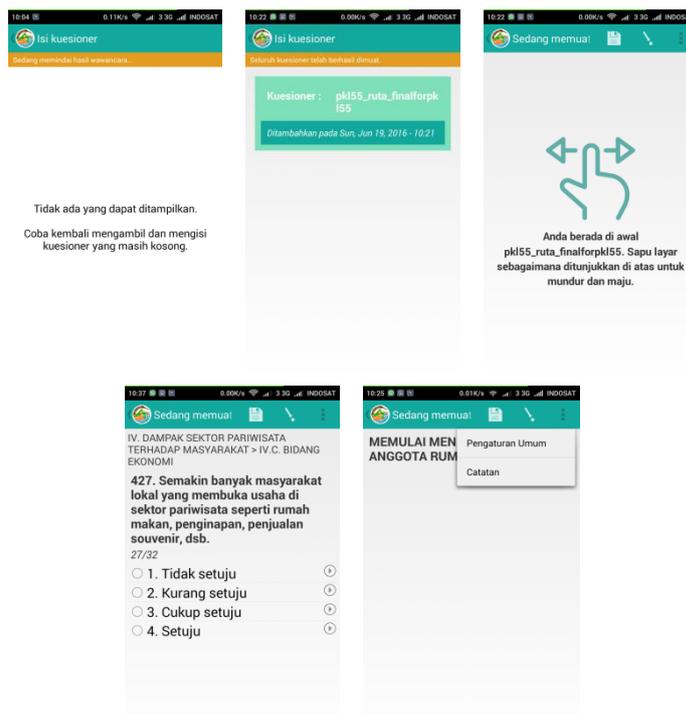
Tampilan aplikasi CAPI PKL 55 yang pertama kali akan muncul ketika pengguna membuka aplikasi adalah *splashscreen* yang menampilkan logo serta *tagline* dari PKL 55. Setelah beberapa saat akan muncul halaman *login* bagi pengguna. Pengguna harus memasukkan *nim* serta *password* agar bisa menggunakan aplikasi CAPI 55.



Gambar 27. Tampilan aplikasi CAPI setelah pengguna *Login*



Setelah pengguna melakukan *login*, selanjutnya pengguna akan memasuki halaman utama atau beranda. Pengguna CAPI 55 dapat terotentikasi sebagai “Kortim” atau “Pencacah”. Perbedaan dua pengguna ini adalah fungsi yang disediakan CAPI 55 untuk keduanya. Kortim akan mempunyai menu monitor sedangkan untuk Pencacah menu monitor tidak disediakan melainkan terdapat menu notifikasi.

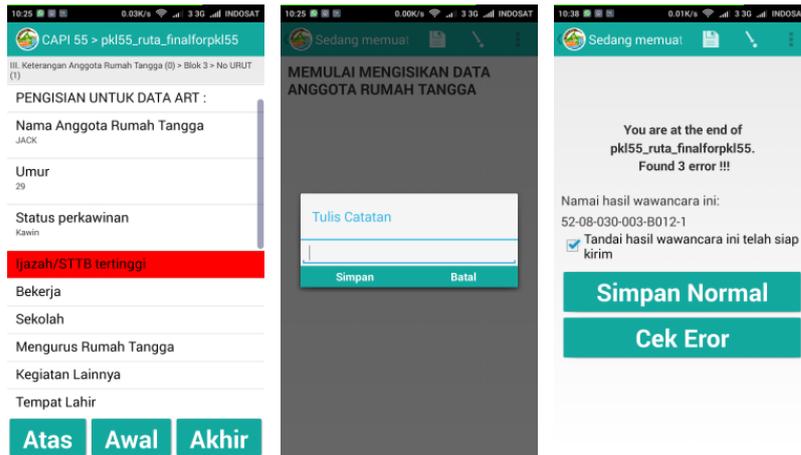


Gambar 28. Tampilan menu isi kuesioner pada aplikasi CAPI

Pada memilih menu Isi Kuesioner, pertama kali kita akan diberikan pilihan untuk memilih kuesioner mana yang akan diisi. Apabila kita belum pernah mengunduh kuesioner sebelumnya, maka tidak ada kuesioner yang ditampilkan.

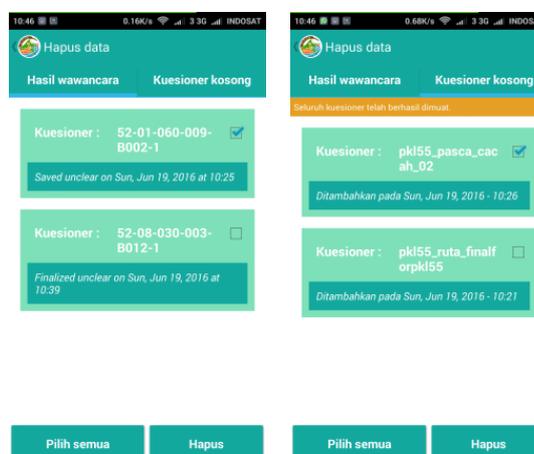
Setelah memilih kuesioner, halaman awal akan menampilkan instruksi ringkas terkait kuesioner. Dalam isi kuesioner sendiri, terdapat beberapa fungsi yang diletakkan di bagian atas. Fungsi-fungsi tersebut diantaranya adalah Simpan, Navigasi, Pengaturan Umum, dan Catatan





Gambar 29. Tampilan fungsi-fungsi pada aplikasi CAPI

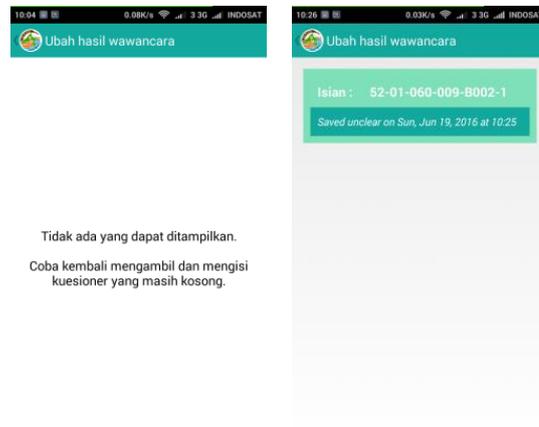
Pada fungsi navigasi, kita bisa memilih pertanyaan mana yang kita tuju. Selain itu disini kita juga bisa melihat isian mana yang mengandung kesalahan dengan tanda merah di barisnya. Ketika fungsi catatan digunakan, akan muncul *pop up* berupa beberapa baris teks untuk pengguna menyisipkan catatan ketika melakukan wawancara dengan responden. Di bagian akhir kuesioner, terdapat halaman yang menampilkan fungsi apakah isian kuesioner ini akan disimpan normal serta siap kirim dan fungsi untuk cek kesalahan.



Gambar 30. Tampilan menu Hapus Data pada aplikasi CAPI



Pada menu Hapus Data, kita bisa menghapus data hasil wawancara atau data kuesioner kosong. Untuk memilihnya cukup klik tab pada bagian atas yang menunjukkan data hasil wawancara atau data kuesioner kosong. Setelah itu berikan tanda cek untuk data yang akan dihapus, kemudian klik hapus.



Gambar 31. Tampilan Ubah Hasil Wawancara pada aplikasi CAPI

Apabila kita belum pernah menyimpan hasil wawancara, maka terdapat tampilan bahwa kita belum pernah mengisi kuesioner kosong. Sedangkan apabila kita sudah pernah mengisi kuesioner sebelumnya dan menyimpannya, maka data tersebut akan muncul. Setelah itu pengguna tinggal memilih data hasil wawancara mana yang akan diubah isianya.

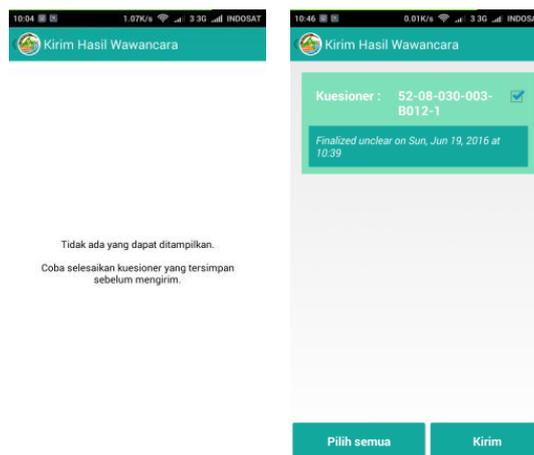


Gambar 32. Tampilan *Download* Kuesioner pada aplikasi CAPI





Pada menu *download* kuesioner, kita bisa memilih beberapa kuesioner yang akan diunduh dengan cara memberi tanda cek pada samping judul kuesioner. Atau kita bisa menekan tombol pilih semua untuk memberi tanda cek pada semua judul kuesioner yang disediakan. Terdapat tombol memuat ulang untuk *refresh* halaman *download* kuesioner. Setelah selesai memberikan tanda cek, maka tekan tombol ambil untuk mengunduhnya.



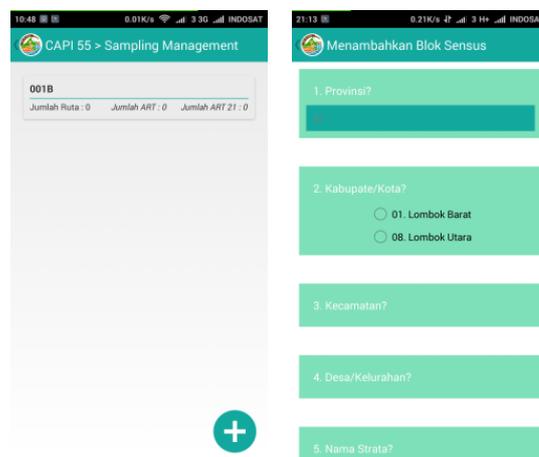
Gambar 33. Tampilan Kirim Hasil Wawancara pada aplikasi CAPI

Hasil wawancara yang telah ditandai untuk siap kirim, selanjutnya akan terlihat pada list data hasil wawancara pada menu kirim hasil wawancara. Apabila belum ada hasil wawancara yang ditandai untuk siap kirim, maka tidak ada data yang ditampilkan pada menu ini. Berikan tanda cek pada data hasil wawancara yang akan dikirim. Setelah selesai memberikan tanda cek, tekan tombol kirim untuk mengirim ke server.



Gambar 34. Tampilan Monitoring pada aplikasi CAPI

Bagi pengguna sebagai kortim, daftar isian hasil wawancara yang dilakukan oleh pencacahnya dapat dilihat pada menu monitoring. Terdapat beberapa status pada isian oleh pencacah yang menandakan apakah isian tersebut sudah tervalidasi benar atau belum. Apabila kortim ingin mengembalikan isian ke pencacah, kortim bisa mengirim ulang ke pencacah dan hasil dari kirim ulang ini akan muncul di menu notifikasi milik pencacah.



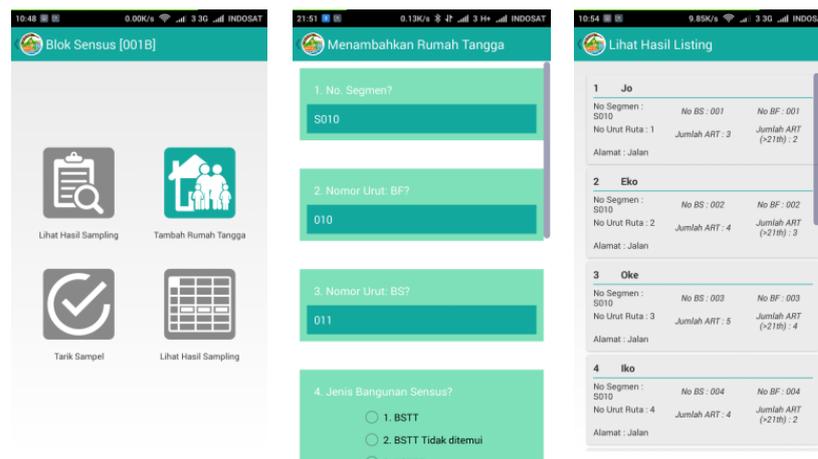
Gambar 35. Tampilan *Sampling Management* pada aplikasi CAPI

Pada menu *Sampling Management*, halaman awal yang ditampilkan adalah list blok sensus yang telah ditambahkan. Apabila pengguna belum pernah



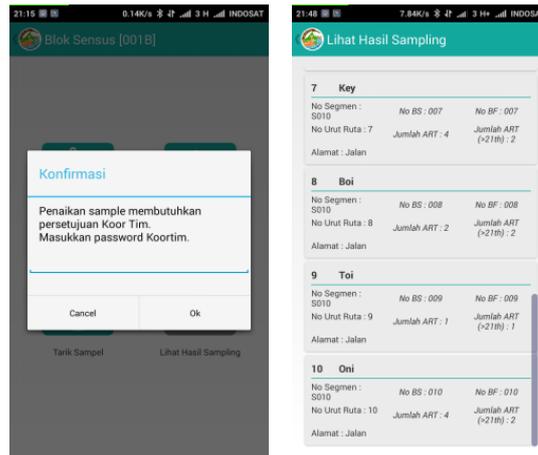


menambahkan blok sensus sama sekali, maka list blok sensus tersebut akan kosong. Untuk menambah blok sensus tekan tombol tanda positif di kiri bawah layar. Selanjutnya akan ditampilkan halaman pengisian blok sensus. Setelah selesai mengisi isian blok sensus, pengguna akan kembali lagi ke halaman list blok sensus. Tekan baris pada list blok sensus untuk masuk ke blok sensus yang akan diambil sampelnya.



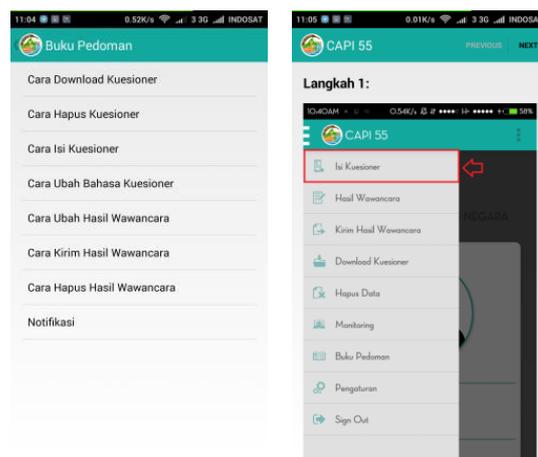
Gambar 36. Tampilan dalam Proses *Sampling* pada aplikasi CAPI

Setelah masuk ke halaman blok sensus, terdapat beberapa tombol di dalamnya. Pada awalnya, tombol yang hanya bisa digunakan adalah tombol tambah rumah tangga. Pengguna bisa menambahkan rumah tangga dengan mengisi isian rumah tangga. Setelah itu tombol lihat hasil listing akan menyala. Pengguna bisa melihat daftar rumah tangga yang telah ditambahkan. Apabila terdapat paling sedikit sepuluh rumah tangga yang telah ditambahkan, tombol tarik sampel akan menyala dan dapat digunakan untuk menarik sampel.



Gambar 37. Tampilan dalam proses konfirmasi kortim pada aplikasi CAPI

Ketika pengguna menekan tombol tarik sampel, maka terdapat perintah untuk memasukkan *password* supervisor (kortim), selanjutnya proses penarikan sampel akan berjalan. Setelah selesai, tombol lihat hasil sampling akan menyala dan pengguna dapat melihat daftar rumah tangga mana saja yang terpilih menjadi sampel.



Gambar 38. Tampilan cara mengunduh buku pedoman pada aplikasi CAPI

Pada menu buku pedoman, pengguna bisa memilih beberapa pedoman terkait beberapa fungsi yang disediakan dalam aplikasi CAPI 55. Pengguna tinggal menekan salah satu baris pada judul pedoman. Selanjutnya pengguna akan





dihadapkan pada halaman yang berisi pedoman berupa foto langkah-langkah penggunaan dari awal hingga akhir. Terdapat tombol arah panah di atas untuk berganti halaman atau pengguna bisa mengusap layar ke kanan atau ke kiri.



Gambar 39. Tampilan pengaturan pada aplikasi CAPI

Pengguna bisa mengatur beberapa konfigurasi terhadap aplikasi CAPI 55 yang telah terpasang. Terdapat beberapa pilihan pengaturan, pengguna tinggal memilih atau menekan baris pada pilihan pengaturan yang telah diberikan.

3.4 *Sampling Management*

Metode Penarikan Sampel *Sampling Management*

Dari desain sampling yang terbentuk oleh seksi Metodologi, pengambilan sampel rumah tangga (ruta) untuk PKL 55 dilakukan secara sistematis sirkuler, dengan sampel *eligible* adalah ruta yang memiliki minimal 1 anggota rumah tangga (ART) berusia lebih dari 21 tahun.

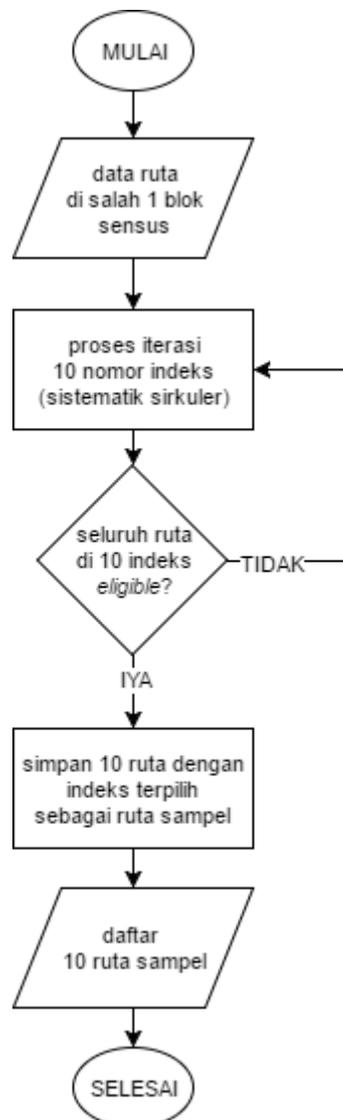
Secara teknis, penarikan sampel dilakukan per blok sensus, dengan ukuran sampel yang disepakati adalah sejumlah 10 ruta *eligible* per blok sensus. Maka





dari itu, implementasi pada *Sampling Management*, pengambilan sampel dilakukan per blok sensus.

Flowchart sampling



Gambar 40. *Flowchart* Desain *Sampling* untuk populasi rumah tangga

Pada blok sensus yang dipilih PCL melalui *interface* CAPI 55, akan terdapat menu “tarik sampel” yang terproteksi *password* yang diketahui oleh kortim saja. Sehingga proses *sampling* hanya bisa dilakukan dengan persetujuan





kortim. Jika *password* sampling diterima, maka akan dilakukan proses sampling sistematis sirkuler pada daftar ruta yang berada di blok sensus bersangkutan. Algoritma yang diimplementasikan pada *Sampling Management* CAPI 55 adalah dengan iterasi 10 nomor indeks secara sistematis. Di mana setiap menemui data ruta, berindeks salah satu dari sepuluh indeks terpilih, dan ruta tersebut tidak memiliki ART yang berusia lebih dari 21 tahun, maka dilakukan pemilihan sepuluh indeks ulang secara sistematis lagi sampai ditemukan sepuluh data ruta *eligible* dengan indeks terpilih.

Berikut kami sertakan sebagian *source code* dari algoritma yang digunakan, yakni bagian fungsi yang bertugas memilih sepuluh indeks. Fungsi di bawah ini dikombinasikan dengan fungsi lain yang bertugas memeriksa syarat *eligible* di setiap data ruta pada nomor indeks terpilih.

```
//FUNGSI pengambilan index sample sistematis sirkuler
// N = jumlah ruta pada frame
// n = jumlah sampel yang dikehendaki
public int[] sistematis(int N, int n) {
    int xint;
    int[] sampel = new int[n];
    boolean ulangi;
    do{
        ulangi = false;
        double i = (double)N/n;
        Random rand = new Random();
        int ar = 0;
        while (ar == 0) {
            ar = rand.nextInt(N);
        }
        int count = 0;
```





```
while (count < n) {  
    //rumus umum sistematik  
    double x = ar + ((count+1)-1)*i;  
    //jika index terpilih > total jumlah unit (N)  
    if(x>N){  
        x = x - N;  
    }  
    //parse x ke integer  
    Double objx = new Double(x);  
    int intx = objx.intValue();  
    //pembulatan  
    if (x == (intx + 0.5) ) {  
        if (intx%2==0) {  
            xint = intx;  
        }  
        else {  
            xint = intx+1;  
        }  
    }  
    else if (x > (intx + 0.5) ) {  
        xint = intx + 1;  
    }  
    else{  
        xint = intx;  
    }  
    //cek jika index 0  
    if(xint==0){  
        ulangi = true;  
        break;  
    }  
    sampel[count] = xint;  
    count++;  
} //while
```





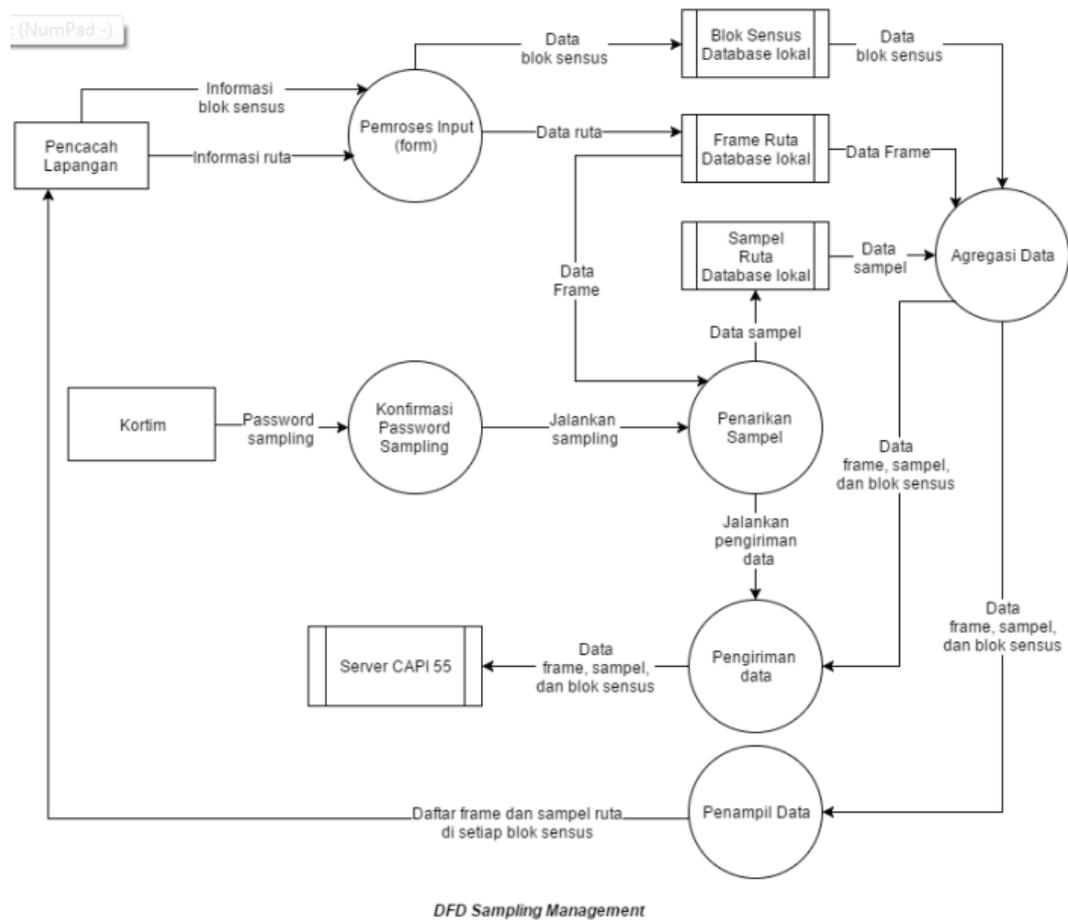
```

}while(ulangi);
return sampel;
}

```

Data Flow Diagram (DFD)

Berikut disajikan *Data Flow Diagram* yang digunakan pada PKL 55. Sistematika yang digunakan mengacu pada kerangka sampling oleh seksi Metodologi dan Listing.

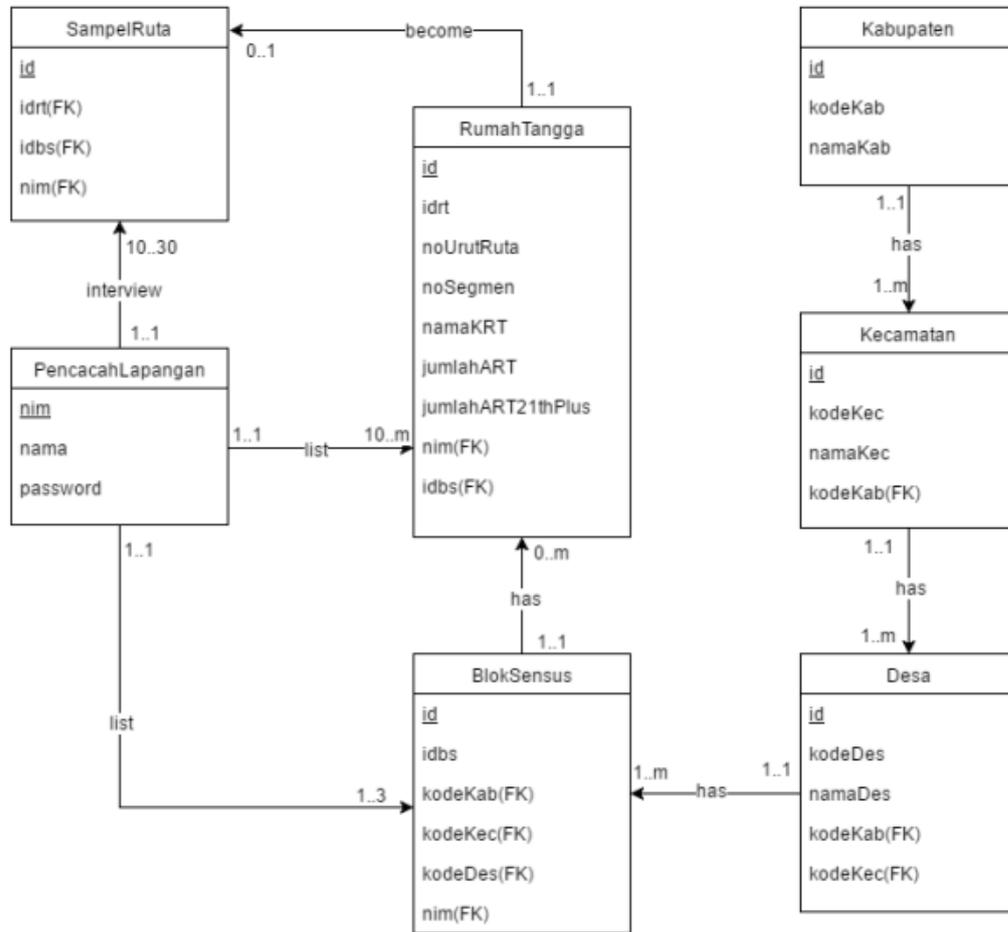


Gambar 41. *Data Flow Diagram Sampling Management*



Entity Relationship Diagram

Berikut disajikan *Entity Relationship Diagram* yang digunakan dalam PKL 55. Sistematika yang digunakan mengacu pada kuesioner pencacahan.



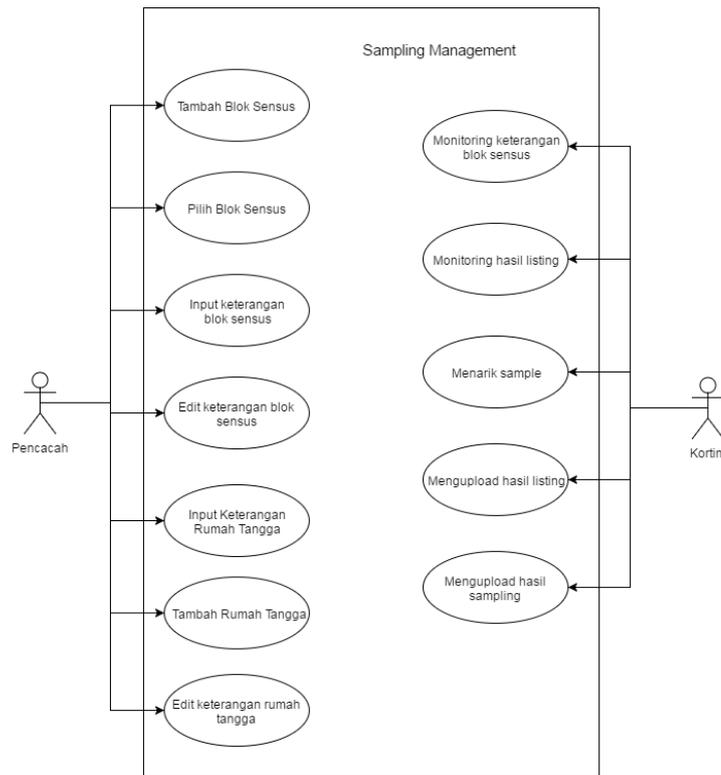
ERD Sampling Management

Gambar 42. *Entity Relationship Diagram Sampling Management*





Use Case

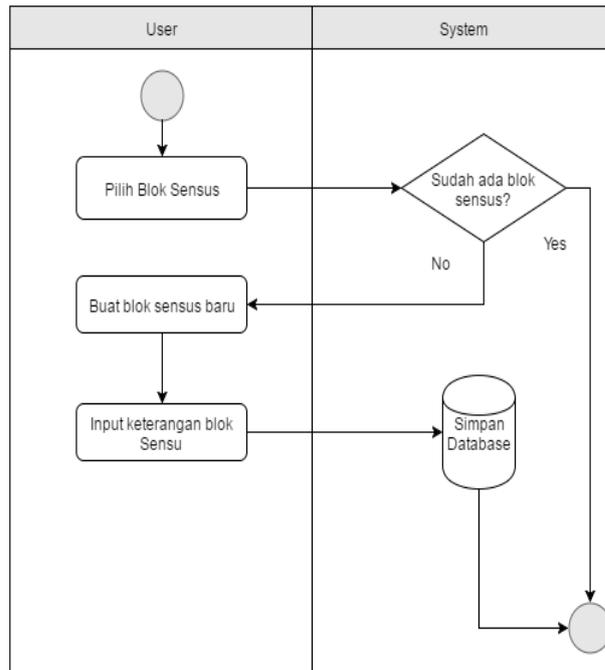


Gambar 43. Use Case pada *Sampling Management*

Dalam sistem *sampling management* yang dirancang pada CAPI 55 terdapat beberapa interaksi antara pencacah dan kortim yang digambarkan melalui use case diagram. Pada *sampling management* CAPI 55, tugas pencacah ialah menginput keterangan blok sensus dan menginput keterangan rumah tangga dalam kegiatan listing, sementara kortim bertugas untuk mengambil sampel dengan cara memasukkan *password* yang telah diberikan kepada kortim. Ketika *password* yang diinputkan oleh kortim cocok maka system akan melakukan tiga hal, yaitu menarik sepuluh sampel dari hasil listing dengan metode SRS, mengupload semua hasil listing ke server, dan mengupload hasil dari penarikan sampel tadi.

Activity Diagram

1. Pilih Blok Sensus

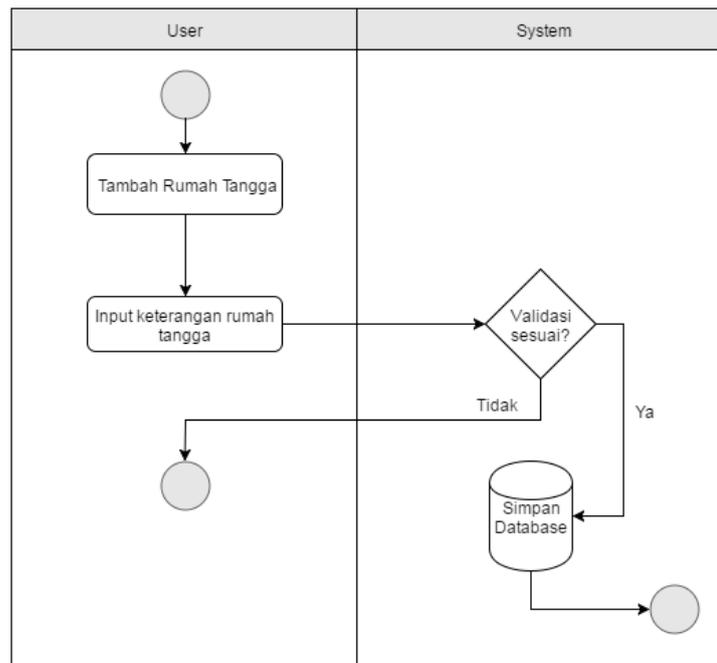


Gambar 44. *Flowchart* Pemilihan Blok Sensus pada *Sampling Management*

Pada saat memilih blok sensus, apabila belum ada blok sensus yang telah dibuat, maka pencacah perlu menambahkan blok sensus baru. Kemudian pencacah menginputkan keterangan mengenai blok sensus. Data keterangan tersebut selanjutnya disimpan pada database SQLite di aplikasi CAPI 55.

2. Isi Keterangan Rumah Tangga

Pada tahap ini pencacah menginputkan keterangan-keterangan rumah tangga dari proses listing. Pada beberapa inputan ada validasi yang harus dipenuhi, apabila salah satu saja validasi tidak terpenuhi maka keterangan rumah tangga tersebut tidak dapat disimpan. Setelah validasi sesuai keterangan-keterangan rumah tangga tersebut disimpan dalam database SQLite pada aplikasi CAPI 55.



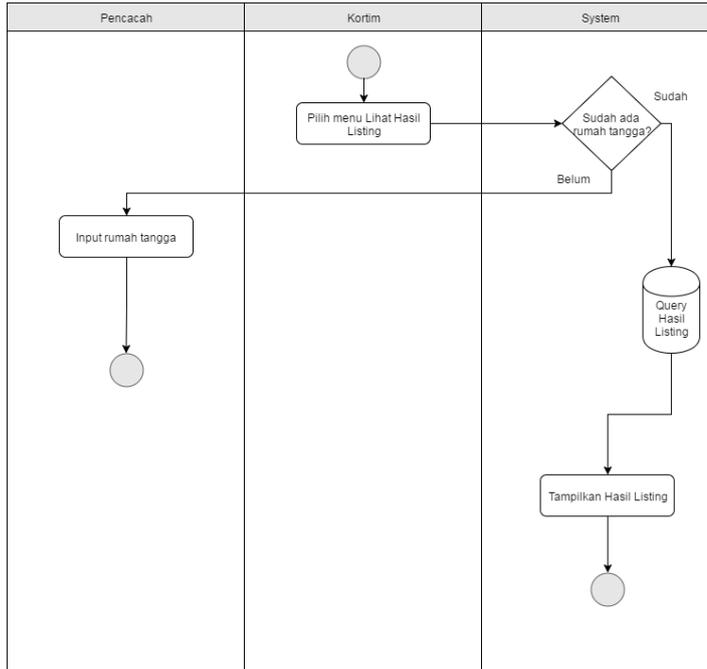
Gambar 45. *Flowchart* dalam pengisian Keterangan Rumah Tangga

Validasi yang ada pada form pengisian keterangan-keterangan rumah tangga di antaranya:

1. Nomor segmen harus 4 digit dan sesuai dengan format nomor segmen,
2. Nomor urut bangunan fisik (BF) harus 3 digit,
3. Nomor urut bangunan sensus (BS) harus 3 digit,
4. Alamat harus diinputkan dengan format yang benar,
5. Anggota rumah tangga 21+ harus lebih kecil sama dengan total anggota keluarga.

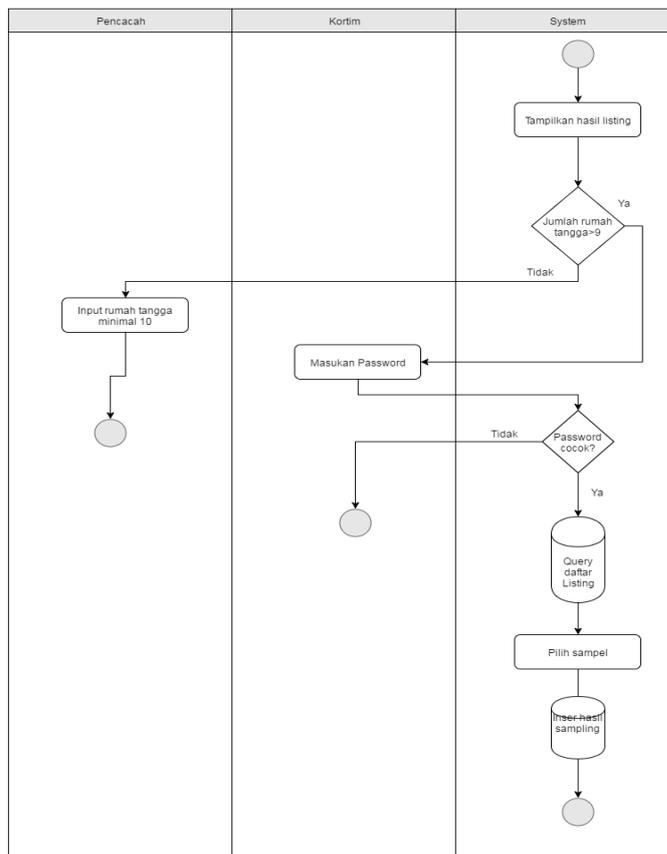
3. Lihat Hasil Listing

Untuk melihat hasil listing, aplikasi CAPI 55 melakukan *query database* terhadap data yang disimpan pada database SQLite. Lalu menampilkan daftar rumah tangga lengkap dengan keterangan-keterangannya.



Gambar 46. *Flowchart* dalam Proses Listing

4. Tarik Sampel

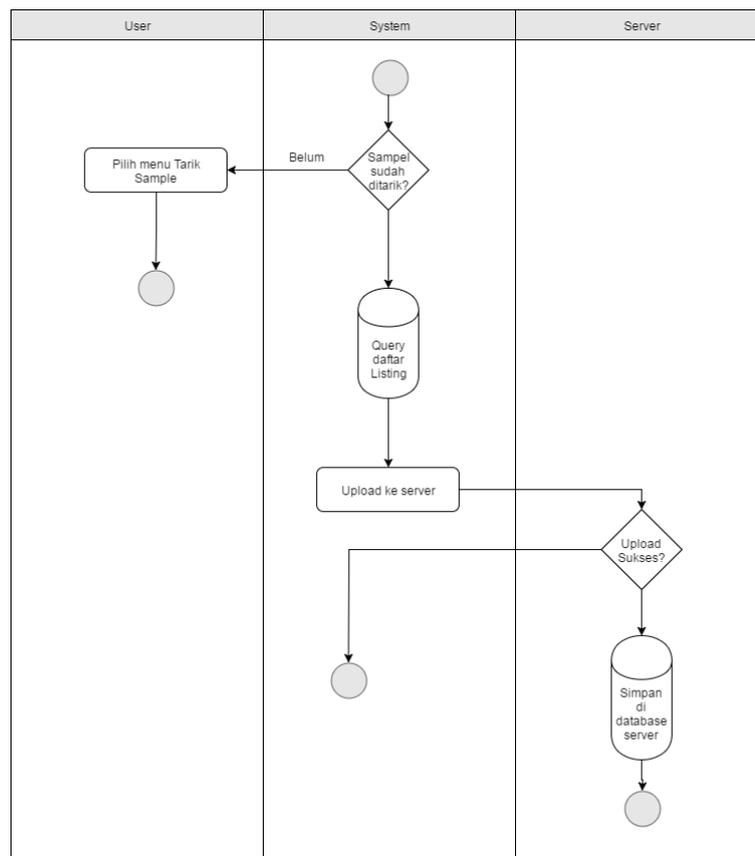


Gambar 47. *Flowchart* Penarikan Sampel pada *Sampling Management*





Pada *sampling management* CAPI 55 digunakan metode penarikan sebanyak sepuluh sample dengan metode SRS. Pada aplikasi CAPI 55 penarikan sampel hanya dapat dilakukan oleh kortim, yaitu dengan cara kortim memasukan *password* yang telah diberikan, setelah itu dengan menggunakan algoritma penarikan sample SRS, sistem akan secara otomatis memilih sampel sesuai dengan kaidah SRS. Setelah itu hasil sampling diinputkan ke dalam database SQLite pada aplikasi CAPI 55



Gambar 48. *Flowchart* Penyimpanan Hasil Penarikan Sampel

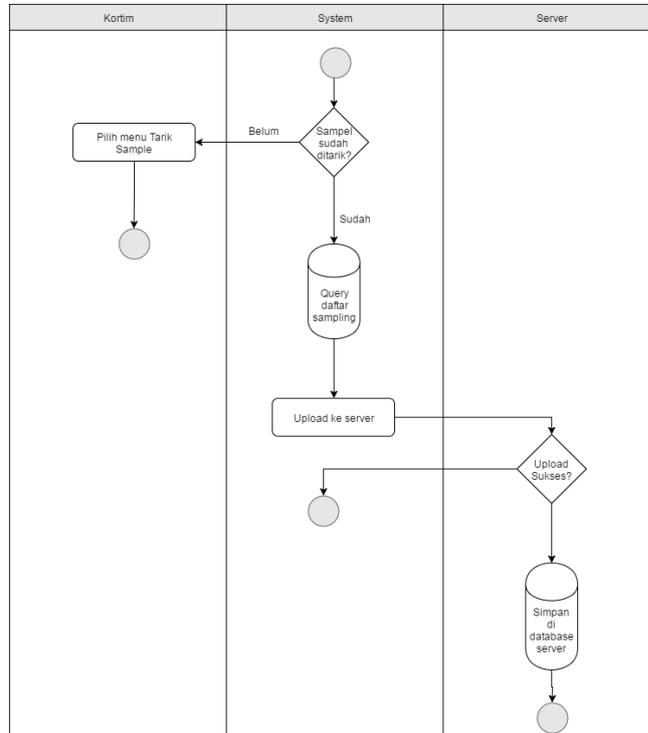
5. Upload Hasil Listing

Proses *upload* hasil listing dilakukan secara otomatis ketika kortim melakukan penarikan sampel dengan cara memasukan *password*. Apabila *upload*



sukses maka daftar listing akan di-*insert* ke database deserver, sedangkan apabila *upload* gagal, maka data masih tersimpan didatabase di aplikasi CAPI 55.

6. Upload Hasil Sampling



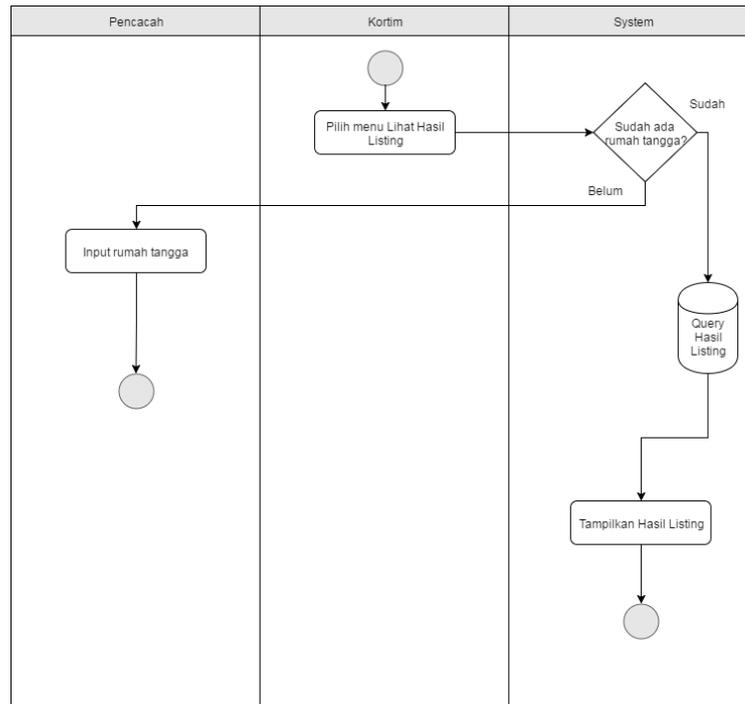
Gambar 49. *Flowchart* dalam proses *Upload Hasil Sampling*

Proses *upload* hasil sampling dilakukan secara otomatis ketika kortim melakukan penarikan sampel dengan cara memasukkan *password*. Apabila *upload* sukses maka daftar *sampling* akan di-*insert* ke database diserver, sedangkan apabila *upload* gagal, maka data masih tersimpan didatabase di aplikasi CAPI 55.

7. Lihat Hasil Sampling

Untuk melihat hasil sampel, maka sistem melakukan *query* terhadap database SQLite yang ada pada aplikasi CAPI 55. Sistem akan menampilkannya lengkap dengan keterangan-keterangan rumah tangga yang dibutuhkan.





Gambar 50. *Flowchart* dalam Menampilkan Hasil Listing



Halaman ini sengaja dikosongkan



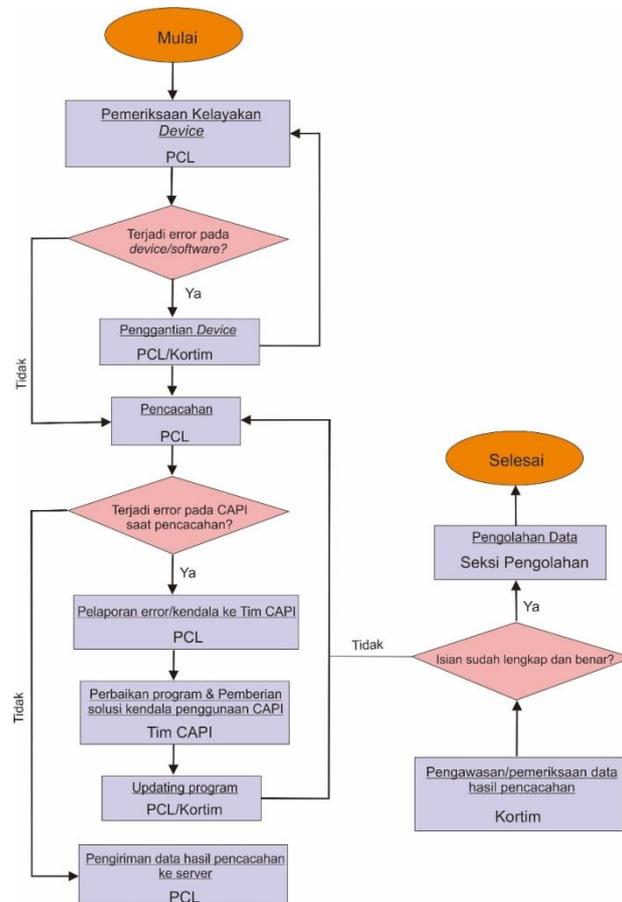


BAB IV

IMPLEMENTASI SISTEM (*Standard Operational Procedure*)

4.1 Deskripsi Penggunaan CAPI

Dalam menjalankan pencacahan lapangan menggunakan CAPI, PCL harus menjalankan tugasnya sesuai SOP pencacahan CAPI yang telah ditentukan. Sehingga diharapkan nantinya akan menghasilkan data yang baik dan berkualitas.



Gambar 51. Alur Pencacahan menggunakan CAPI

Dalam pelaksanaan lapangannya, *Standard Operational Procedure* secara umum dapat dijabarkan sebagai berikut:





- a. PCL mengisi baterai *tablet* hingga *full* pada malam hari sebelum hari pencacahan,
- b. menghemat daya *tablet* pada saat pencacahan dengan cara;
 - 1) mematikan *tablet* jika belum dipakai. Misalnya, jika sedang dalam perjalanan;
 - 2) mematikan koneksi data jika tidak diperlukan,
 - 3) dan lain-lain.
- c. PCL tidak mengeluarkan *tablet* jika pencacahan belum dimulai dan belum bertemu responden,
- d. tidak memainkan *tablet* saat melakukan pencacahan,
- e. mengunggah *file backup* CAPI ke *directory Google Drive* yang telah ditentukan.

4.2 Profil Pengguna CAPI

Dalam menggunakan CAPI terdapat dua subjek, yaitu Kortim dan PCL. Kortim bertugas dalam mengecek dan mengawasi PCL mereka masing-masing saat pencacahan dilapangan. Setelah isian pencacahan sudah lengkap dan benar, kortim mengirim hasil pencacahan ke server. Jika masih belum, maka isian tersebut akan dikembalikan ke masing-masing PCL untuk dilakukan pencacahan ulang.

PCL berperan penting dalam melakukan pencacahan. Mereka melakukan listing di lapangan, menarik sampel, dan mengisi isian saat pencacahan. Setelah isian sudah lengkap dan tidak ada *error*, maka PCL mengirim isian kuesionernya





ke server. Selanjutnya kortim akan melakukan pengecekan terhadap masing-masing isian kuesioner yang sudah dikirim.

4.3 Penjelasan per Fungsi

Login

Pada tampilan pertama aplikasi CAPI akan muncul tampilan logo CAPI dan *login*. Tampilan *login* terdapat dua isian. Pertama, isian NIM, silahkan mengisi NIM sesuai dengan NIM anda yang dimiliki. Kedua, isian *password*, silahkan mengisi *password* yang telah diberikan (*default*: (NIM)).



Gambar 52. Tampilan awal *login* pada Aplikasi CAPI

Contoh:

NIM : 13.5555

Password : "password sesuai dengan sipadu PKL"

Jika *Password* sudah diubah, silahkan mengisi *password* yang sudah diubah. Pastikan NIM dan *Password* anda benar. Setelah NIM dan *Password* diisi,



silahkan memencet tombol *Sign In* dibawah isian *login*. Jika NIM dan *Password* Anda benar, maka Anda akan dibawa ke tampilan awal pencacahan.

Isi Kuesioner

Tombol menu ini berisi kuesioner yang akan digunakan saat melakukan pencacahan.

Hasil wawancara

Tombol menu ini berisi hasil dari wawancara setelah melakukan pencacahan.

Kirim hasil wawancara

Tombol menu ini berfungsi untuk mengirim hasil wawancara ke kortim.

Download kuesioner

Tombol menu ini berfungsi untuk mendownload kuesioner-kuesioner yang diperlukan untuk pencacahan.

Hapus kuesioner

Tombol menu ini berfungsi untuk menghapus kuesioner

Monitoring

Tombol menu berfungsi yang ada di Kortim untuk melihat seluruh hasil wawancara PCL dibawah Kortim. Beberapa kondisi yang bisa dipantau adalah jumlah responden yang sudah di cacah dan hasil wawancara yang sudah di





unggah. Jika memilih menu monitoring akan muncul tampilan seperti disamping. Pada menu ini, kortim dapat mengunduh dan melakukan pengecekan terhadap hasil wawancara dari PCL dengan menekan hasil wawancara.

Notifikasi

Tombol menu ini berisi notifikasi kuesioner yang dikembalikan.

Buku Pedoman

Tombol menu ini berisi panduan cara penggunaan CAPI.

Pengaturan

Tombol menu ini berisi pengaturan.

Sign out

Tombol menu ini untuk keluar dari aplikasi CAPI.



Halaman ini sengaja dikosongkan





BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Pembuatan aplikasi CAPI dilakukan agar dapat membantu proses pencacahan dan pengolahan data sehingga dapat berjalan dengan efisien dan efektif. Untuk merancang aplikasi CAPI dibutuhkan *Workflow*, *Bussiness Process*, *Data Flow Diagram (DFD)*, *Entity Relationship Diagram (ERD)*, *Use Case*, dan *Graphical User Interface*.
2. Dalam CAPI PKL 55 dapat mengakomodir pencacahn dan kegiatan sampling. Subsistem yang terdapat CAPI PKL 55 adalah build, *aggregate*, collect dan *sampling management*.
3. Aplikasi CAPI yang dibangun pada PKL 55 adalah CAPI yang bersifat dinamis sehingga tidak hanya dapat digunakan untuk satu form kuesioner saja tetapi juga dapat digunakan untuk kuesioner lain dan aplikasi CAPI ini dapat digunakan kembali pada kegiatan pencacahan di luar PKL 55.

5.2 Saran

1. Alur dari pembuatan dan penggunaan CAPI lebih efisien dan efektif. Oleh karena itu, CAPI dapat mulai digunakan untuk pertimbangan dalam proses pencacahan dibandingkan dengan PAPI.





2. *Hardware* dan *software* berpengaruh secara langsung terhadap pencacah yang menggunakan metode CAPI, sehingga pencacahan selanjutnya lebih memerhatikan alur penanganan masalah.
3. Dokumentasi sistem dilakukan secara komprehensif dalam perekaman alur kerja dan pembangunan sistem serta pelaksanaan di lapangan dan ditingkatkan kembali.
4. Penggunaan *hardware* sebagai media CAPI yang cocok, yaitu:
 - a. ukuran standar *tablet* 10,1 inci,
 - b. ukuran *pixel* minimum sebesar 1280 x 800 *pixels* dan dimensi 262.6 x 175.4 x 8.6 mm,
 - c. berat antara 280 - 560 gr, dan
 - d. jenis baterai *Lithium polymer battery 2-cell* berkapasitas 6000 mAh yang memiliki ketahanan baterai sampai 24 jam.
5. Perlu diadakan penelitian lebih lanjut demi menyempurnakan sistem CAPI pada PKL selanjutnya dengan harapan aplikasi CAPI mampu menjadi alternatif dalam pengumpulan data yang lebih optimal baik dari segi kualitas, biaya, maupun waktu.





PUBLIKASI PKL 55

SEKOLAH TINGGI ILMU STATISTIK

Jalan Otto Iskandardinata No. 64c Jakarta 13330
Telp. (021) 8191437, 8508812 | Fax : (021)8197577