SISTEM ANTREAN DENGAN PEMANGGILAN NAMA PADA PEMUNGUTAN SUARA MENGGUNAKAN RFID

SKRIPSI

Diajukan Untuk Memenuhi Sebagian Persyaratan

Memperoleh Gelar Sarjana Teknik

DISUSUN OLEH:

ASHRI LINTANG AS.

NIM. 0910630005-63

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
JURUSAN TEKNIK ELEKTRO
MALANG
2013

repository.ub.ac.id

SISTEM ANTREAN DENGAN PEMANGGILAN NAMA PADA PEMUNGUTAN SUARA MENGGUNAKAN RFID

SKRIPSI

Diajukan Untuk Memenuhi Sebagian Persyaratan

Memperoleh Gelar Sarjana Teknik

DISUSUN OLEH:

ASHRI LINTANG AS.
NIM. 0910630005-63

Telah diperiksa dan disetujui oleh
Dosen Pembimbing:

Pembimbing I	**Pembimbing II**
Mochammad Rif'an, ST., MT.	**Zainul Abidin, ST., MT., M. Eng.**
NIP. 19710301 200012 1 001	**NIK. 86012306110279**

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

LEMBAR PENGESAHAN

SISTEM ANTREAN DENGAN PEMANGGILAN NAMA PADA PEMUNGUTAN SUARA MENGGUNAKAN RFID

SKRIPSI
JURUSAN TEKNIK ELEKTRO

Diajukan untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik

Disusun oleh:
ASHRI LINTANG AS.
NIM. 0910630005-63

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 22 Juli 2013

DOSEN PENGUJI

Ir. Ponco Siwindarto, M. Eng. Sc
NIP. 19590304 198903 1 001

Ir. M. Julius St., M.S.
NIP. 19540720 198203 1 002

Ir. Nurussa'adah, M.T.
NIP. 19680706 199203 2 001

Mengetahui
Ketua Jurusan Teknik Elektro

Dr. Ir. Sholeh Hadi Pramono, M.S.
NIP. 19580728 198701 1 001

UNIVERSITAS BRAWIJAYA

PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT, karena dengan rahmat, taufik dan hidayah-Nya lah skripsi ini dapat diselesaikan.

Skripsi berjudul "Sistem Antrean dengan Pemanggilan Nama pada Pemungutan Suara Menggunakan RFID" ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

- Allah SWT yang telah melimpahkan kesehatan, kemudahan serta kelancaran dalam mengerjakan skripsi ini hingga akhir,
- Ibu Alfiah Udayani dan almarhum Bapak Wimbo Karno Rusetiyo Budi atas segala doa, nasehat, kasih sayang, perhatian dan kesabarannya dalam membesarkan dan mendidik penulis, serta telah banyak memberikan motivasi dan mendoakan kelancaran penulis hingga terselesaikannya skripsi ini,
- Keluargaku semua, Adikku Adek Kharisma Rezal Fajrin yang telah membantu memberikan masukan dalam menyelesaikan skripsi ini,
- Dr. Ir. Sholeh Hadi Pramono, M.S. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya,
- M. Aziz Muslim, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
- Ir. Nurussa'adah, MT. selaku Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya,
- Mochammad Rif'an, ST., MT. dan Zainul Abidin, ST., MT., M.Eng. selaku Dosen Pembimbing 1 dan Dosen Pembimbing 2 atas segala bimbingan, nasehat, pengarahan, motivasi, saran dan masukan yang telah diberikan,
- Staf rekording Jurusan Teknik Elektro,
- Kharisma Cahaya Aqli dan Tri Evanti Andriani, sahabat seperjuangan, teman berbagi dan teman cerita penulis sejak awal perkuliahan hingga kini, serta Dwisnita K. (Bona), teman seperjuangan selama menempuh perkuliahan konsentrasi Elektronika hingga skripsi ini terselesaikan,

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

- Teman – teman galauers, Nur Iskandar Juang, partner dalam mengerjakan tugas – tugas kelompok perkuliahan, Eky Prasetya, Raditya Artha R., Wahyu Suwito, teman bermain dan belajar yang baik selama menjalani perkuliahan konsentrasi Elektronika,
- Teman – teman asisten Laboratorium Komputasi dan Jaringan, mas Diriga dan mas Putra Prima yang telah membantu dalam menangani masalah dalam skripsi ini, Maulani Candra, Mas Jumadil dan Mas Felix yang telah menjadi teman seperjuangan penulis untuk menjadi asisten, Firman, Soraya, Krisna, Fahad, Fahmy, Liza, Abdur dan asisten – asisten senior yang telah memberikan support dalam menyelesaikan skripsi ini,
- Dahniar Pepy L, Firda A, Yoga S, Sam Budi S., Gladi Buana dan teman – teman el-Kampank lain yang telah menjadi keluarga baru pada awal perkuliahan penulis,
- Keluarga besar Cumad yang telah memberikan motivasi untuk segera menyelesaikan skripsi ini,
- Teman – teman di kelembagaan Workshop periode 2012/2013, B – Lovers, tim LTC angkatan 2009, Ampere, Panphas serta semua pihak yang tidak mungkin bagi penulis untuk mencantumkan namanya satu – persatu, terima kasih banyak atas bantuan dan dukungannya.

Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Malang, Juli 2013

Penulis

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRAK

Ashri Lintang Aldi Safitri, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Juli 2013, *Sistem Antrean dengan Pemanggilan Nama pada Pemungutan Suara Menggunakan RFID*, Dosen Pembimbing: Mochammad Rif'an, ST., MT. dan Zainul Abidin, ST., MT., M.Eng.

Sistem antrean otomatis berbasis teknologi dengan pemanggilan nomor antrean telah dimanfaatkan oleh beberapa fasilitas layanan publik. Sistem otomatis ini memiliki kelemahan yakni adanya kewajiban bagi pengantre untuk mengingat nomor antrean agar tidak harus mengantre kembali dari awal ketika antreannya terlewati. Skripsi ini menjelaskan tentang sistem antrean dengan pemanggilan nama yang ditujukan untuk kegiatan pemungutan suara. Peserta pemungutan suara (*voter*) diidentifikasi melalui tag RFID (*Radio Frequency Identification*) pasif TK4100 yang dibaca oleh RFID *Starter Kit* berbasis ID-12. Pemanggilan antrean *voter* dilakukan secara visual dan audio dengan mengonversi data tulisan menjadi data suara melalui *software* Borland Delphi 7 dengan sumber data berasal dari *database* yang dibuat menggunakan MySQL. Dari hasil pengujian didapatkan bahwa jarak baca maksimal RFID *Starter Kit* adalah 4,5 cm. Hasil pengujian juga menunjukkan bahwa sistem mampu melakukan pencarian data *voter*, penyimpanan urutan antrean, pemanggilan antrean serta pengubahan status *voter* yang telah melakukan pemungutan suara dengan tingkat keberhasilan 100%.

Kata Kunci : Sistem Antrean Pemungutan Suara, Tag RFID TK4100, RFID *Starter Kit*, *text to speech*.

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGANTAR i
ABSTRAK iii
DAFTAR ISI iv
DAFTAR GAMBAR vi
DAFTAR TABEL ix
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Batasan Masalah 3
1.4 Tujuan 3
1.5 Sistematika Penulisan 4
BAB II TINJAUAN PUSTAKA 5
2.1 Teori Dasar RFID 6
2.1.1 Tag RFID 7
2.1.2 RFID *Reader* 9
2.1.3 RFID *Starter Kit* 10
2.1.4 Tag RFID TK4100 12
2.2 Komunikasi Serial 13
2.2.1 RS232 13
2.3 *Database* 14
2.3.1 MySQL 17
2.4 Delphi 7 18
2.5 Microsoft SAPI (*Speech Application Programming Interface*) 18
BAB III METODOLOGI PENELITIAN 19

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

3.1 Perancangan Alat 19
3.2 Pembuatan Alat 19
3.3 Pengujian Alat 19
BAB IV PERANCANGAN DAN PEMBUATAN 22
4.1 Penentuan Spesifikasi Alat 22
4.2 Perancangan Alat 23
4.2.1 Perancangan Subrutin Pembacaan Tag RFID oleh RFID *Starter Kit* 26
4.2.2 Perancangan Subrutin Pencarian Data *Voter* dalam *Database* 26
4.2.3 Perancangan Subrutin Pemanggilan Antrean *Voter* 27
4.2.4 Perancangan *Database* 28
4.3 Pembuatan Alat 30
4.3.1 Pengaturan RFID *Starter Kit* 30
4.3.2 Pembuatan *Software* 30
BAB V PENGUJIAN DAN ANALISIS 36
5.1 Pengujian Jarak Pembacaan Tag RFID oleh RFID *Starter Kit* 36
5.2 Pengujian Pencarian Data *Voter* dalam *Database* dan Menyimpan Data dalam Urutan Antrean 38
5.3 Pengujian Pemanggilan Urutan Antrean *Voter* Menurut Nomor Bilik 41
5.4 Pengujian Perubahan Status Pemungutan Suara *Voter* 43
5.5 Pengujian Sistem Secara Keseluruhan 47
BAB VI KESIMPULAN DAN SARAN 52
6.1 Kesimpulan 52
6.2 Saran 53
DAFTAR PUSTAKA 54

DAFTAR GAMBAR

Gambar 2.1 Contoh RFID *reader* dan tag RFID 6
Gambar 2.2 Hubungan komponen – komponen pada sistem RFID 7
Gambar 2.3 Macam – macam bentuk dan ukuran tag RFID 8
Gambar 2.4 Bagian – bagian tag RFID 8
Gambar 2.5 Tata letak dan *setting jumper* RFID *Starter Kit* 10
Gambar 2.6 *Setting jumper* untuk pemilihan format data 11
Gambar 2.7 Gambaran bagian J1 tampak depan 12
Gambar 2.8 Tag RFID TK4100 12
Gambar 2.9 Port DB9 dan DB25 jenis *female* 14
Gambar 2.10 RS232 Serial UART *Convertor* 14
Gambar 2.11 Komponen – komponen utama sistem basis data 16
Gambar 4.1 Diagram Blok Sistem 23
Gambar 4.2 *Flowchart* perancangan perangkat lunak sistem secara keseluruhan 25
Gambar 4.3 *Flowchart* perancangan subrutin pembacaan tag RFID oleh RFID *Starter Kit* 26
Gambar 4.4 *Flowchart* perancangan subrutin pencarian data *voter* dalam *database* 27
Gambar 4.5 *Flowchart* perancangan subrutin pemanggilan antrean *voter* 28
Gambar 4.6 *Setting jumper* RFID mode *reader only* dengan format data UART RS-232 (ASCII) 30
Gambar 4.7 Diagram Blok program tampilan pada PC 31
Gambar 4.8 Program tampilan utama 32
Gambar 4.9 Tampilan program pengisian nama kegiatan 32
Gambar 4.10 Tampilan program identifikasi *voter* 33
Gambar 4.11 Tampilan program identifikasi *voter* yang kehilangan tag RFID 33
Gambar 4.12 Tampilan program pengontrol pemanggilan *voter* 34

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 4.12 (a) Hasil pembuatan tabel utama data *voter* pada *database* (b) Hasil pembuatan tabel antrean *voter* pada *database* 35

Gambar 5.1 Tag RFID yang digunakan untuk pengujian pertama 37

Gambar 5.2 Tampilan hasil pengujian pertama 37

Gambar 5.3 *Flowchart* pengujian pencarian data dan penyimpanan data dalam antrean 38

Gambar 5.4 Tabel utama *database voter* 39

Gambar 5.5 Hasil pengujian pertama pencarian data *voter* 39

Gambar 5.6 Hasil pengecekan tabel *database* antrean untuk pengujian pertama 39

Gambar 5.7 Tabel *database* antrean hasil penyimpanan urutan antrean *voter* 40

Gambar 5.8 *Flowchart* pengujian pemanggilan urutan antrean *voter* menurut nomor bilik 41

Gambar 5.9 Tabel *database* antrean yang digunakan sebagai acuan pemanggilan 42

Gambar 5.10 Tampilan visual hasil pengujian pemanggilan *voter* yang pertama 42

Gambar 5.11 *Flowchart* pengujian perubahan status pemungutan suara *voter* 44

Gambar 5.12 Tabel *database* antrean yang digunakan sebagai acuan pengujian perubahan status pemungutan suara *voter* 44

Gambar 5.13 Tampilan pemanggilan *voter* yang pertama secara visual sebelum dilakukan simulasi pemungutan suara 45

Gambar 5.14 Tampilan program pengecekan status pemungutan suara *voter* yang pertama 45

Gambar 5.15 Tampilan kotak dialog hasil pengecekan perubahan status pemungutan suara *voter* yang pertama 46

Gambar 5.16 Tabel *database* antrean setelah dilakukan simulasi pemungutan suara *voter* 46

Gambar 5.17 (a) Tampilan pengujian pengisian nama TPS dan pengecekan password sistem (b) Tampilan pengujian pengisian nama kegiatan pemungutan suara yang dilakukan 47

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 5.18 Tampilan pengidentifikasian data *voter* yang pertama setelah dimasukkan ke dalam antrean 48

Gambar 5.19 Tampilan pemanggilan antrean *voter* secara visual yang pertama..... 48

Gambar 5.20 (a) Tampilan pengidentifikasian data *voter* kedua (b) Tampilan pengidentifikasian data *voter* ketiga (c) Tampilan pengidentifikasian data *voter* keempat dengan kartu tag hilang (d) Tampilan pengidentifikasian data *voter* kelima (e) Tampilan pengidentifikasian data voter keenam 50

Gambar 5.21 Tabel *database* antrean setelah dilakukan pemanggilan antrean secara keseluruhan..... 51

DAFTAR TABEL

Tabel 2.1 Perbandingan tag aktif, pasif dan semi pasif 9

Tabel 2.2 Konfigurasi hubungan antara COM port DB9 dengan RFID *Starter Kit* 12

Tabel 2.3 Persamaan fungsi pin DB9 dan DB25 14

Tabel 4.1 Atribut – atribut tabel utama *database* 29

Tabel 4.2 Atribut – atribut tabel antrean *database* 29

Tabel 5.1 Hasil pengujian jarak pembacaan tag RFID oleh RFID *Starter Kit* 37

Tabel 5.2 Hasil pengujian pencarian data *voter* dalam *database* 39

Tabel 5.3 Hasil pengujian pemanggilan urutan antrean *voter* menurut nomor bilik 42

Tabel 5.4 Hasil pengujian perubahan status pemungutan suara *voter* 46

Tabel 5.5 Hasil pengujian keseluruhan pemanggilan *voter* secara audio dan visual 50

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemungutan suara merupakan bagian dari sistem demokrasi. Kegiatan ini dilakukan untuk menentukan orang – orang yang berhak untuk menduduki suatu jabatan politik tertentu di suatu wilayah. Salah satu negara yang menganut sistem demokrasi dan melaksanakan kegiatan pemungutan suara secara rutin adalah negara Indonesia.

Pemungutan suara di Indonesia kini dilakukan mulai dari tingkat desa (pemilihan Kepala Desa), Kota atau Kabupaten (pemilihan Walikota/Bupati dan anggota DPRD tingkat 2), provinsi (pemilihan Gubernur dan anggota DPRD tingkat 1), hingga tingkat pemerintah pusat (Presiden dan anggota DPR). Pemungutan suara dilaksanakan untuk suatu periode tertentu yang biasanya dalam hitungan tahun. Misalnya saja pemungutan suara untuk pemilihan Presiden yang dilakukan setiap 5 tahun sekali. (Undang – Undang Republik Indonesia Nomor 23 Tahun 2003 tentang Pemilihan Umum Presiden dan Wakil Presiden Pasal 3 Ayat 2, 2003)

Pemungutan suara atau biasa disebut dengan pemilihan umum (pemilu) merupakan suatu kegiatan yang melibatkan orang banyak. Orang – orang yang terlibat dalam kegiatan ini tidak hanya terbatas pada kalangan politik saja, namun meliputi seluruh masyarakat yang telah memenuhi syarat. Di Indonesia, salah satu syarat untuk memperoleh hak memberikan suara dalam pemilu adalah terdaftar sebagai warga negara dengan usia 17 tahun atau lebih. (Undang – Undang Republik Indonesia Nomor 23 Tahun 2003 tentang Pemilihan Umum Presiden dan Wakil Presiden Pasal 7, 2003)

Menurut data yang dimiliki Komisi Pemilihan Umum pada pemilu Presiden Indonesia tahun 2009, tercatat bahwa jumlah orang yang memiliki hak untuk memilih (*voter*) sekitar 170 juta orang. Jumlah ini tentunya sangat besar apabila dibandingkan dengan fasilitas TPS (Tempat Pemungutan Suara) yang

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

hanya berjumlah sekitar 500 ribu TPS. Karena keterbatasan jumlah TPS dalam menangani jumlah *voter*, maka pada saat pelaksanaan pemungutan suara biasanya terjadi antrean *voter*. Metode antrean *voter* ini diatur sendiri oleh masing – masing TPS. Metode yang paling sering digunakan ada 2 macam, yakni menggunakan metode geser duduk (*voter* bergeser tempat duduk sesuai urutan) dan menggunakan pemanggilan nama oleh panitia pemilu. Kedua metode ini bila dibandingkan dengan sistem antrean berbasis teknologi yang telah dikembangkan saat ini, termasuk metode yang sangat manual.

Sistem antrean berbasis teknologi yang ada saat ini adalah antrean dengan pemanggilan nomor urutan otomatis. Sistem antrean ini dapat ditemui pada pelayanan teller bank atau pelayanan kantor pos. Setiap orang yang ingin mengantre akan diberi secarik kertas berisi nomor antrean yang akan dipanggil ketika tiba giliran. Kelemahan dari sistem dengan pemanggilan nomor urutan ini adalah pengantre diharuskan untuk mengingat nomor antreannya. Ketika pengantre lupa akan nomor urutannya dan pemanggilan nomor urutannya telah terlewati, maka pengantre harus mengambil nomor antrean ulang.

Dengan melihat kelemahan yang ada pada sistem antrean pemanggilan nomor, maka apabila sistem ini akan diterapkan pada antrean pemungutan suara tentunya harus ada pengembangan lebih lanjut. Hal ini disebabkan karena mengingat *voter* yang akan mengantre meliputi seluruh kalangan masyarakat yang belum tentu semuanya mengerti akan metode ini. Oleh karena itu, pada tugas akhir ini akan dirancang sebuah sistem antrean pada pemungutan suara yang diusahakan mampu menangani kelemahan yang ada pada sistem antrean pemanggilan nomor. Sistem ini akan menggabungkan sistem antrean pemanggilan nomor dengan metode pemanggilan nama manual yang ada pada sistem antrean pemilu sebelumnya. Diharapkan sistem antrean ini dapat dimanfaatkan sebagai acuan agar dapat dikembangkan dalam berbagai aplikasi.

1.2 Rumusan Masalah

Mengacu pada permasalahan yang disebutkan dalam latar belakang, maka ditetapkan rumusan masalah sebagai berikut :

1) Bagaimana merancang dan membuat *software* antarmuka *Personal Computer* dengan RFID *Starter Kit*.
2) Bagaimana merancang dan membuat *software* PC yang dapat menampilkan dan menyimpan data *voter* dalam suatu urutan antrean.
3) Bagaimana merancang dan membuat *software* PC yang dapat memanggil *voter* secara berurutan.
4) Bagaimana merancang dan membuat *software* PC yang dapat memberikan tanda untuk *voter* yang telah melakukan pemungutan suara.

1.3 Batasan Masalah

Karena luasnya objek kajian maka perlu dilakukan pembatasan masalah agar pembahasan lebih terfokus pada rumusan masalah. Adapun batasan masalah dalam makalah ini adalah:

1) Sistem antrean yang dirancang merupakan sistem antrean untuk pemungutan suara dengan jumlah antrean/*user* maksimal 999 orang.
2) Alat identifikasi *voter* menggunakan tag RFID dan RFID *Starter Kit*.
3) *Database voter* didapatkan langsung dari pengisian data secara manual oleh penulis.
4) *Voter* yang dapat melakukan pemilihan dianggap telah memiliki tag RFID yang telah didaftarkan pada *database*.
5) Sistem antrean pemungutan suara hanya meliputi cara memasukkan *voter* ke dalam antrean serta cara untuk memanggil antrean *voter*.
6) Bilik suara yang disediakan hanya berjumlah 3 buah.
7) Pemanggilan nama *voter* dibuat menggunakan fungsi *text to speech* dan *database* bahasa Inggris buatan Microsoft (Microsoft SAPI).

1.4 Tujuan

Tujuan penelitian ini adalah untuk merancang sebuah sistem pengatur antrean menggunakan pemanggilan nama *voter* pada pemungutan suara dengan identifikasi RFID yang memberikan kemudahan bagi *voter* dalam melakukan antrean.

1.5 Sistematika Penulisan

Untuk memberikan gambaran secara garis besar guna memudahkan pembaca dalam memahami penelitian ini, sistematika penulisan yang digunakan adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, batasan permasalahan, tujuan, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menguraikan tentang teori – teori yang mendukung perancangan Sistem Antrean dengan Pemanggilan Nama pada Pemungutan Suara Menggunakan RFID.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang langkah – langkah yang dipakai untuk menyelesaikan masalah dalam perancangan, pengujian dan analisis data.

BAB IV PERANCANGAN DAN PEMBUATAN

Bab ini menjelaskan tentang perancangan dan pembuatan alat yang meliputi spesifikasi, perencanaan diagram blok, prinsip kerja dan realisasi sistem.

BAB V PENGUJIAN DAN ANALISIS

Bab ini memuat penjelasan tentang cara pengujian serta data hasil pengujian yang dilakukan. Data hasil pengujian ini kemudian akan dianalisis demi kelayakan sistem dan keberhasilan sistem untuk mencapai tujuan perancangan. Pengujian dan analisa akan dilakukan secara per blok dan secara keseluruhan sistem.

BAB VI KESIMPULAN DAN SARAN

Bab ini menguraikan tentang kesimpulan dari sistem yang telah dirancang serta saran untuk pengembangan sistem selanjutnya.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB II

TINJAUAN PUSTAKA

Sistem antrean dengan pemanggilan nomor urutan mulai digunakan secara luas pada fasilitas – fasilitas pelayanan publik di Indonesia. Contoh penggunaan yang paling sering ditemui adalah pada pelayanan teller bank. Setiap nasabah yang ingin melakukan transaksi akan diberi secarik kertas yang berisi nomor antrean. Beberapa bank ada yang telah menggunakan alat khusus yang akan mencetak kertas antrian ketika nasabah menginginkan untuk masuk ke dalam antrean.

Salah satu sistem antrean dengan pemanggilan nomor yang pernah dirancang adalah milik Maulina Wulandari, mahasiswi Teknik Elektro Universitas Brawijaya Malang yang dirancang pada tahun 2011. Sistem antrean tersebut diaplikasikan untuk antrean pada apotek dengan menggunakan *barcode* sebagai alat identifikasi. Pemanggilan antrean dilakukan dengan cara memanggil nomor antrean melalui *speaker* dan *seven segment*.

Setelah mempelajari sistem antrean dan sistem pemungutan suara yang telah ada, maka pada perancangan tugas akhir ini akan dilakukan beberapa pengembangan sesuai dengan tujuan sistem. Karena sistem yang direncanakan adalah sistem antrean pada pemungutan suara, maka pada perancangan ini akan ditambahkan dengan pengenalan identitas *voter* untuk pengambilan nomor antrean. Selain pemanggilan nomor antrean seperti pada sistem yang telah ada, akan ditambahkan pula pemanggilan nama *voter* sebagai pengembangan sistem. Pemanggilan nama dimaksudkan untuk menghindari keteledoran *voter* karena lupa akan nomor antrean.

Dalam merencanakan dan merealisasikan sistem antrean dengan pemanggilan nama, maka dibutuhkan pemahaman tentang berbagai hal yang mendukung. Pemahaman dalam tinjauan pustaka ini mengacu latar belakang yang dapat mendukung perencanaan alat. Pada bagian ini mendukung teori tentang

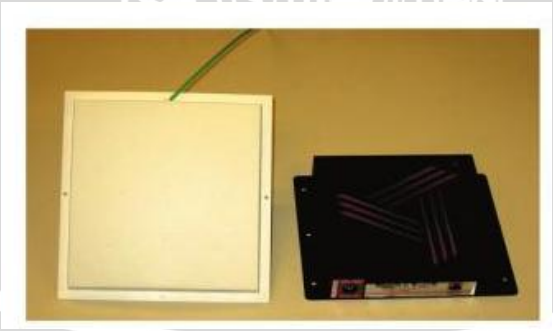
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

bagaimana merancang alat yang dapat mengidentifikasi *voter* menggunakan tag RFID dan RFID *Starter Kit*, mencari data dalam *database*, pemanggilan data melalui *speaker*, dan dapat mengeluarkan tampilan dalam monitor. Berdasarkan hal tersebut, tinjauan pustaka dalam makalah ini meliputi kajian RFID, *database*, komunikasi serial, Delphi dan Microsoft SAPI.

2.1 Teori Dasar RFID

Radio Frequency Identification atau lebih dikenal dengan singkatan RFID adalah teknologi identifikasi berbasis gelombang radio (*Radio Frequency*) yang mampu mengidentifikasi berbagai objek secara simultan tanpa diperlukan kontak langsung. RFID menggunakan sarana yang disebut label RFID atau transponder untuk menyimpan dan mengambil data jarak jauh.

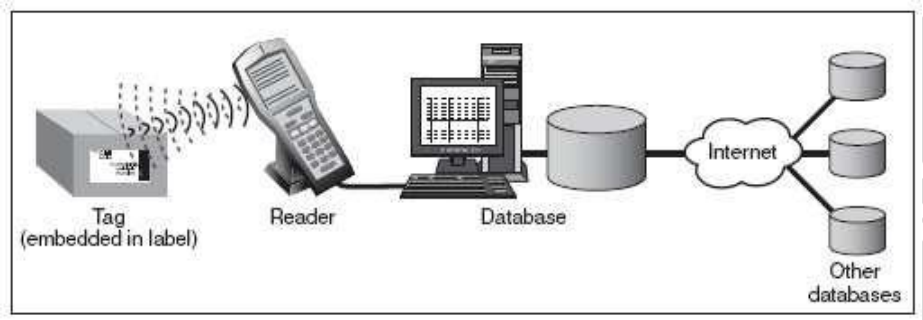
Sistem RFID secara utuh memiliki tiga komponen dasar yang terdiri dari tag RFID atau kadang disebut transponder, RFID *reader* atau bisa disebut introgator dan kontroler atau host. Tag RFID berfungsi sebagai tempat penyimpanan data identitas dari objek yang diikuti. Introgator bertanggungjawab untuk membaca data dari tag RFID dan kemudian dikirimkan menuju kontroler. Kontroler merupakan otak dari sistem RFID yang bertugas sebagai pusat pemrosesan informasi. Kontroler pada beberapa jaringan umumnya adalah komputer (PC) atau *workstation* yang menjalankan *database* atau aplikasi tertentu. Contoh tag RFID dan RFID *reader* dapat ditunjukkan dalam Gambar 2.1.



Gambar 2.1 Contoh RFID *reader* dan tag RFID
Sumber : Roy Want, 2006.

Tag RFID dan introgator akan bertukar informasi melalui gelombang radio dengan frekuensi tertentu. Ketika suatu objek memasuki area pembacaan dari introgator, introgator akan memberikan sinyal kepada tag RFID untuk

mengirimkan data. Data yang dimaksud adalah sekumpulan informasi mengenai objek, biasanya berupa nomor seri objek, yang disimpan di dalam tag RFID. Data yang diterima oleh introgator selanjutnya akan segera dikirimkan menuju kontroler melalui antarmuka standar. Kontroler atau host komputer merupakan sistem yang akan mengatur alur informasi dari item – item yang terdeteksi dalam lingkup sistem RFID dan mengatur komunikasi antara tag dan *reader*. Host dapat berupa komputer *stand – alone* atau komputer yang terhubung ke jaringan LAN atau internet untuk berkomunikasi dengan server. Gambar 2.2 menunjukkan hubungan setiap komponen pada sistem RFID.



Gambar 2.2 Hubungan komponen – komponen pada sistem RFID
Sumber : Winda, 2009.

Teknologi RFID memiliki beberapa keunggulan dibandingkan dengan teknologi identifikasi yang lain. Keunggulan RFID adalah pada aspek efisiensi dan kenyamanan penggunaannya. Tag RFID mampu diidentifikasi secara simultan (bersamaan) oleh RFID *reader* tanpa harus berada dalam jarak dekat. Selain itu, keunggulan lain dari teknologi RFID adalah kemampuan tag RFID yang dapat diidentifikasi menembus berbagai objek seperti kertas, plastik dan kayu.

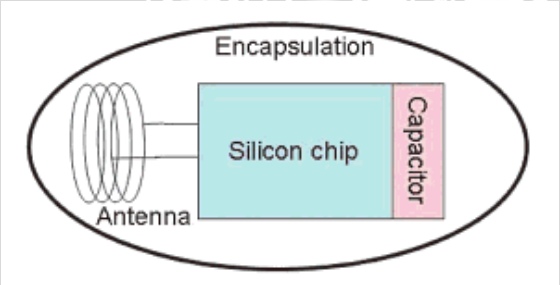
2.1.1 Tag RFID

Tag RFID pada dasarnya merupakan suatu mikrochip berantena yang disertakan pada suatu unit barang. Tag RFID terdiri atas chip semikonduktor berupa mikrochip silikon, antena dan ada pula yang memiliki baterai. Setiap tag RFID memiliki *Electronic Product Code* (EPC) yang berisi identitas dari objek. EPC terdiri atas angka – angka yang menunjukkan kode – kode tertentu, misalnya kode produsen, produk atau nomor seri dari objek yang diikutinya. Ukuran bit EPC yang mencapai 96 bit memungkinkan adanya konfigurasi yang mampu

mengidentifikasi lebih dari 268 juta produsen, masing – masing memiliki lebih dari satu juta jenis produk dan sisanya masih cukup untuk dijadikan label seluruh produk individualnya.

Tag RFID memiliki bentuk dan ukuran yang beragam. Ada yang berbentuk seperti gantungan kunci, ada pula yang seperti kartu identitas biasa. Bentuk ini merupakan bentuk *encapsulated* dari chip dan antena tag. Sebenarnya sebelum "dibungkus" ukuran tag dan antena RFID kecil dan mudah rusak sehingga tidak praktis untuk pemakaian di lapangan. Oleh karena itu dilakukan enkapsulasi yang dapat melindungi tag RFID dari kondisi lingkungan yang ekstrem. Gambar 2.3 menunjukkan beberapa contoh tag RFID yang sudah mulai digunakan, sedangkan Gambar 2.4 menunjukkan gambaran sederhana bagian - bagian tag RFID.

Gambar 2.3 Macam – macam bentuk dan ukuran tag RFID
Sumber : Michael McRoberts, 2010.



Gambar 2.4 Bagian – bagian Tag RFID
Sumber : Roy Want, 2006.

Tag RFID terdiri atas tiga jenis, yakni tag RFID aktif, pasif dan semi pasif. Ketiga jenis tag ini dibedakan dari segi sumber penyediaan energi listrik yang dibutuhkan oleh tag dalam membangkitkan chip.

Sebuah tag aktif membutuhkan baterai kecil yang menyediakan energi listrik untuk membangkitkan dan mentransmisikan sinyal RF (*Radio Frequency*)

secara terus menerus. Tag aktif mampu dibaca oleh RFID *reader* yang diletakkan pada jarak kurang lebih 30 meter. Umumnya tag ini memiliki sejumlah memori yang penting untuk menyimpan informasi. Untuk beberapa kondisi, introgator dapat membaca data dari tag aktif dan kemudian mengirimkan sinyal untuk memberikan kode kembali pada tag dengan informasi atau instruksi yang baru.

Berbeda dengan tag aktif, tag pasif tidak membutuhkan baterai dalam pengoperasiannya. Tag jenis ini mendapatkan energi dari medan elektromagnetik yang dihasilkan oleh *reader*. Ketika tag melewati area medan elektromagnetik dari RFID *reader*, koil yang ada pada antena tag akan terinduksi sehingga muncul arus. Arus yang mengalir digunakan untuk membangkitkan chip kecil yang ada pada tag untuk segera mengirimkan data menuju *reader*. Karena memori yang relatif kecil, tag pasif hanya mampu menyimpan data berupa nomor ID saja. Informasi lengkap lainnya akan disimpan pada perangkat lain atau dapat juga disimpan pada memori eksternal tag.

Jenis tag yang ketiga, yaitu tag RFID semi pasif merupakan gabungan antara tag aktif dan tag pasif. Tag ini memiliki sebuah baterai kecil yang akan di-charge setiap tag memasuki medan elektromagnetik dari *reader*. Perbandingan ketiga jenis tag ditunjukkan dalam Tabel 2.1.

Tabel 2.1 Perbandingan Tag Aktif, Pasif dan Semi Pasif

	Tag Aktif	**Tag Pasif**	**Tag Semi Pasif**
Sumber energy	Baterai	Induksi dari gelombang elektromagnetik yang dipancarkan oleh *reader*	Baterai dan induksi
Jarak pembacaan	Hingga 30 meter	3 – 7 meter	Hingga 30 meter
Ketelitian pembacaan	Buruk	Baik	Buruk
Interferensi frekuensi	Tinggi	Sedang	Tinggi
Media penyimpanan	32 kb atau lebih Read/Write	2 kb Read only	32 kb atau lebih Read/Write

Sumber : Edmund W. Schuster, Stuart J. Allen dan David L Brock, 2007.

2.1.2 RFID *Reader*

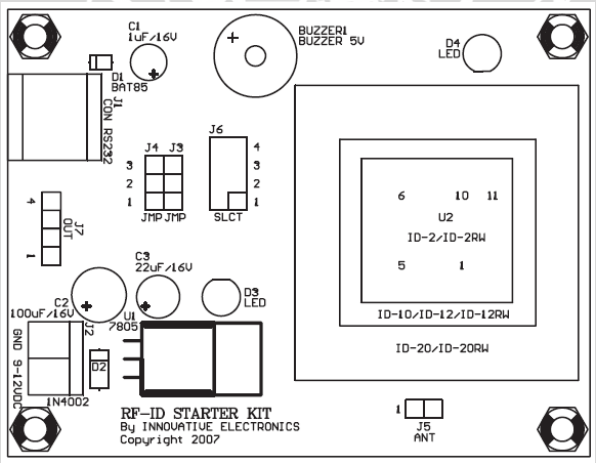
Sebuah RFID *reader* berperan sebagai jembatan penghubung antara RFID tag dan kontroler atau host. RFID *reader* akan membaca informasi yang ada pada

tag RFID melalui gelombang radio dengan frekuensi tertentu. RFID *reader* umumnya memiliki beberapa fungsi dasar, yakni membaca isi data yang ada pada tag RFID, menulis data ke dalam tag RFID (dalam bentuk *smart tag*), mengirim data menuju dan dari kontroler serta memberikan energi ke tag RFID (pada tag RFID pasif).

RFID *reader* atau introgator sebenarnya adalah sebuah komputer kecil yang terdiri atas antena, modul elektronik RF yang bertanggungjawab untuk mengkomunikasikan *reader* dengan tag RFID dan modul kontrol elektronik yang berperan untuk mengkomunikasikan *reader* dengan kontroler.

2.1.3 RFID *Starter Kit*

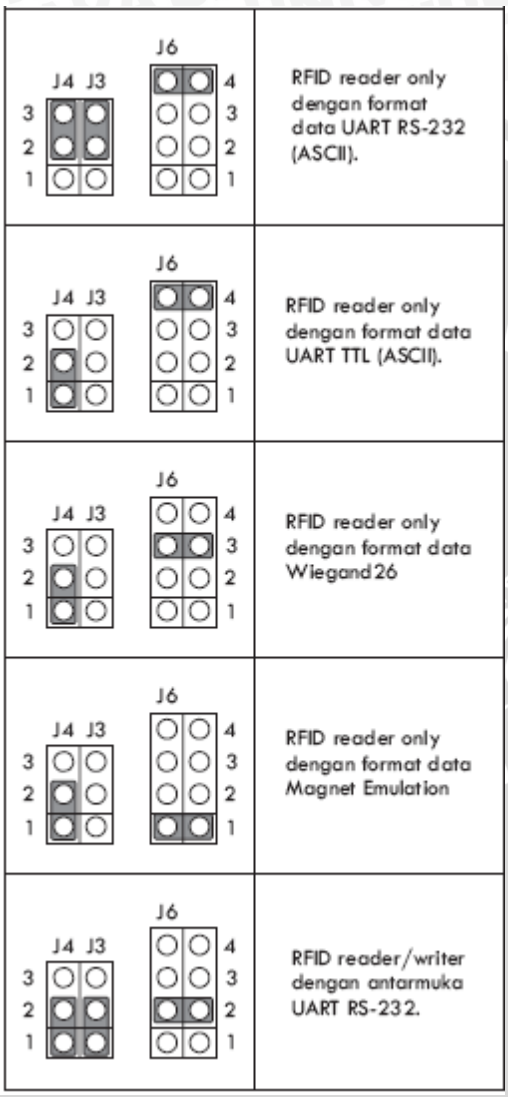
RFID *Starter Kit* merupakan suatu modul yang digunakan sebagai suatu sarana pengembangan RFID berbasis *reader* ID-12 dengan frekuensi kerja 125 kHz. Dengan frekuensi tersebut, modul ini dapat membaca beberapa tipe tag, salah satunya adalah tag yang bertipe EM4001 dengan jarak baca maksimal 12 cm. Gambar 2.5 menunjukkan tata letak dan *setting jumper* dari RFID *Starter Kit*.



Gambar 2.5 Tata Letak dan *Setting jumper* RFID *Starter Kit*
Sumber : Innovative Electronics, 2007.

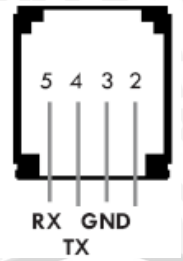
Dalam pengoperasiannya, RFID *Starter Kit* membutuhkan catu daya DC sebesar 9 – 12 V yang dapat dihubungkan melalui J2. Sebagai indikator kerja, modul ini telah dilengkapi dengan 2 buah LED dan sebuah buzzer. LED yang terletak pada bagian D3 berfungsi sebagai indikator suplai daya, LED yang ada pada bagian D4 sebagai indikator tulis, sedangkan buzzer berfungsi sebagai

indikator baca. Modul ini mendukung format data berupa ASCII (UART TTL/RS-232), Wiegand26 maupun Magnetic ABA Track2 (Magnet Emulation) yang dapat diatur melalui pemasangan *jumper* J3, J4 dan J6. *Setting jumper* untuk pemilihan format data dapat ditunjukkan dalam Gambar 2.6.



Gambar 2.6 *Setting jumper* untuk Pemilihan Format Data
Sumber : Innovative Electronics, 2007.

Untuk kepentingan antarmuka dengan komputer, RFID *Starter Kit* telah dilengkapi dengan jalur komunikasi serial UART RS-232 dengan konektor RJ11 yang terletak pada bagian J1. Gambaran tampak depan dari RJ11 dapat dilihat dalam Gambar 2.7. Bagian ini hanya digunakan untuk RFID *reader only* dalam mode UART RS-232 (ASCII) dan RFID *reader*/*writer*. Hubungan antara COM port DB9 dengan RFID *Starter Kit* adalah *straight* dengan konfigurasi yang ditunjukkan dalam Tabel 2.2.



Gambar 2.7 Gambaran Bagian J1 Tampak Depan
Sumber : Innovative Electronics, 2007.

Tabel 2.2 Konfigurasi Hubungan antara COM port DB9 dengan RFID *Starter Kit*

Fungsi	COM port Komputer DB9	RFID *Starter Kit* J1
RX	Pin 2	Pin 5
TX	Pin 3	Pin 4
GND	Pin 5	Pin 3

Sumber : Innovative Electronics, 2007.

2.1.4 Tag RFID TK4100

Tag RFID TK4100 merupakan salah satu jenis tag RFID pasif yang dapat dibaca oleh RFID *reader* tipe ID - 12. Tag yang berdimensi 85,6 mm×53,98 mm×0,84mm ini bekerja pada frekuensi 125 kHz. Tag ini hanya mampu menyimpan data saja tanpa bisa diprogram kembali (*Read only*) dengan kapasitas informasi yang dapat disimpan adalah sebesar 64 bit. Tag ini memperoleh energi dari hasil induksi medan elektromagnetik *reader* yang diterima oleh koil (antena). Sistem enkoding tag ini menggunakan Manchester. Bentuk tag RFID TK4100 ditunjukkan dalam Gambar 2.8.

Gambar 2.8 Tag RFID TK4100
Sumber *: http://indo-ware.com*

2.2 Komunikasi Serial

Komunikasi serial adalah komunikasi yang melakukan pengiriman data secara berurutan bit per bit. Data yang akan dikirim melalui komunikasi ini akan dipecah – pecah menjadi bit – bit pembentuknya agar dapat ditransmisikan. Data yang sudah dipecah kemudian akan disusun kembali menjadi data utuh ketika telah sampai pada tujuan. Komunikasi jenis ini sering digunakan untuk pengiriman data jarak jauh sehingga biasanya menggunakan kabel yang panjang.

Peralatan yang ada pada komunikasi serial dibagi menjadi dua kelompok, yaitu *Data Communication Equipment* (DCE) dan *Data Terminal Equipment* (DTE). Contoh dari DCE adalah modem, plotter dan scanner, sedangkan contoh DTE adalah terminal yang ada di komputer. Spesifikasi elektronik dari serial port merujuk pada *Electronic Industry Association* (EIA) :

1) "Space" (logika 0) adalah tegangan antara +3 hingga +25V
2) "Mark" (logika 1) adalah tegangan antara -3 hingga -25V
3) Daerah antara +3 hingga -3V merupakan daerah yang tidak terdefinisi
4) Tegangan rangkaian terbuka tidak boleh melebihi 25V
5) Arus hubungan singkat tidak boleh melebihi 500 mA

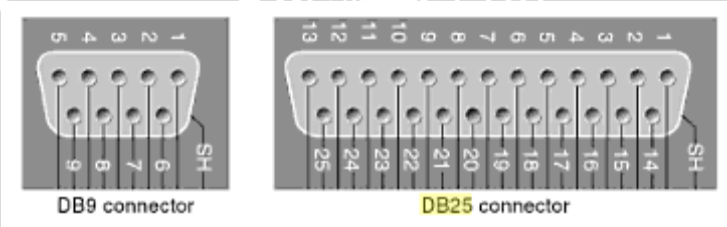
2.2.1 RS232

Standar antarmuka untuk komunikasi serial komputer adalah RS232. RS232 harus disambungkan dengan konektor (port) agar dapat terhubung dengan terminal pada komputer. Konektor yang dapat digunakan ada dua jenis, yaitu port DB9 yang memiliki 9 pin dan port DB25 yang memiliki 25 pin. Masing – masing port ini memiliki pasangan, yaitu port *male* (jantan) yang terdapat pada perangkat dan port *female* (betina) yang terpasang pada komputer. Kedua jenis port, baik DB9 maupun DB25 memiliki beberapa pin dengan fungsi yang sama seperti ditampilkan dalam Tabel 2.3. Gambar 2.9 menunjukkan bentuk port DB9 dan DB25 jenis *female*. Sayangnya, kedua jenis port serial ini sudah mulai jarang tersedia pada komputer – komputer terbaru sehingga untuk menghubungkannya perlu ditambahkan dengan konversi serial port menuju USB. Contoh konverter serial port menuju USB dapat dilihat dalam Gambar 2.10.

Tabel 2.3 Persamaan Fungsi Pin DB9 dan DB25

Pin DB9	Pin DB25	Fungsi
1	8	Data carrier detect
2	3	Receive data
3	2	Transmit data
4	20	Data terminal ready
5	7	Signal ground
6	6	Data set ready
7	4	Request to send
8	5	Clear to send
9	22	Ring indicator

Sumber : Massoud Moussavi, 2012.



Gambar 2.9 Port DB9 dan DB25 jenis female
Sumber : Massoud Moussavi, 2012.

Gambar 2.10 RS232 Serial UART *Converter*
Sumber : Sunrom Technologies, 2012.

2.3 *Database*

Basis data (*database*) adalah kumpulan dari beberapa berkas data atau informasi yang kompleks, yang disusun menjadi beberapa kelompok dengan tipe data yang sejenis di mana setiap data di dalamnya dapat saling terkait satu sama lain atau berdiri sendiri. Kegiatan yang dilakukan dalam basis data umumnya adalah merancang basis data, mengumpulkan berbagai macam data dan mengolah

serta menggali data untuk berbagai keperluan. Aplikasi yang digunakan untuk melakukan operasi basis data ini disebut aplikasi basis data.

Sistem pengelolaan *database* disebut dengan DBMS (*Database Management System*). DBMS sendiri merupakan perangkat yang berfungsi untuk mengelola *database* secara efektif dan efisien. Kebanyakan DBMS menyediakan mekanisme pengaturan sekuritas terhadap basis data berdasarkan wewenang dari pengguna. Sekuritas data sangat penting pada basis data untuk menghindari adanya pengaksesan data secara ilegal.

Dalam menjalankan fungsinya, DBMS memiliki beberapa komponen utama seperti yang ditunjukkan dalam Gambar 2.11. Komponen – komponen utama DBMS ini antara lain :

a) Perangkat keras (*hardware*)

   Perangkat keras merupakan komponen DBMS yang berfungsi untuk melakukan pemrosesan dan menyimpan basis data.

b) Data

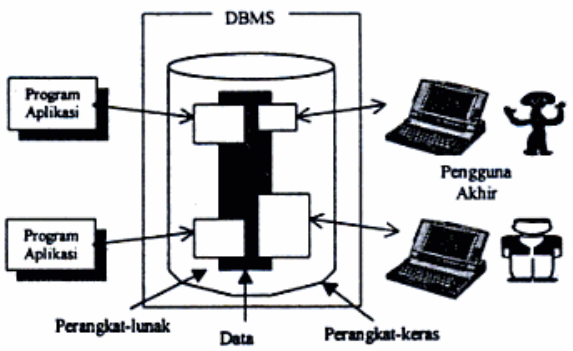
   Data yang ada dalam basis data memiliki sifat terpadu (*integrated*) dan berbagi (*shared*). Terpadu dalam hal ini berarti bahwa berkas – berkas data yang ada pada basis data memiliki keterkaitan satu sama lain sehingga tidak terjadi kemubaziran data. *Shared* atau berbagi berarti bahwa data dapat diakses oleh sejumlah pengguna dalam waktu bersamaan. Sifat ini biasanya dimiliki oleh sistem basis data *multiuser*.

c) Perangkat lunak

   Dalam hal ini DBMS berperan sebagai jembatan antara basis data dan pengguna. Perangkat lunak inilah yang bertugas untuk melayani permintaan – permintaan pengguna dalam mengolah basis data.

d) Pengguna

   Pengguna adalah seseorang yang menyusun sistem basis data untuk tujuan tertentu.



Gambar 2.11 Komponen – komponen utama sistem basis data
Sumber : Abdul Kadir, 1999.

Pada dasarnya DBMS mengenal dua macam bentuk *database*, yakni flat-file dan relational. *Database* flat-file merupakan bentuk *database* yang menyimpan semua data dalam satu tabel, sedangkan *database* relational adalah *database* yang memiliki beberapa tabel yang saling berkaitan. *Database relational* memiliki sebuah komponen yang berfungsi sebagai penghubung antar tabel yang disebut *primary key*. *Primary key* ini tidak boleh memiliki data yang berulang atau kembar. *Database* dengan bentuk *relational* ini lebih dikenal dengan nama RDBMS (*Relational Database Management System*)

RDBMS merepresentasikan data dalam bentuk tabel – tabel yang disusun dalam bentuk baris (*record*) dan kolom (*field*). Beberapa contoh aplikasi yang menggunakan sistem ini di antaranya adalah Oracle, MySQL, Sybase, dBase, Microsoft Access dan MS SQL.

RDBMS memiliki tiga kelompok perintah yang digunakan dalam mengelola dan mengorganisasikan data, yaitu :

a) *Data Definition Language* (DDL)

   Perintah ini digunakan oleh administrator untuk mendefinisikan struktur *database* seperti membuat tabel baru, menentukan struktur penyimpanan tabel, model relasi antar tabel dan validasi data.

b) *Data Manipulation Language* (DML)

   Perintah ini digunakan untuk memanipulasi dan mengambil data pada suatu *database* agar dapat memudahkan pengguna dalam mengakses isi *database*. Perintah yang termasuk dalam DML misalnya adalah

menampilkan data (*Select*), menambah data (*Insert*), menghapus data (*Delete*) dan mengubah data (*Update*).

c) *Data Control Language*

Perintah ini berhubungan dengan cara mengendalikan data yang mengarah pada segi sekuritas data untuk mengatur hak akses seorang pemakai *database*.

2.3.1 MySQL

MySQL merupakan salah satu sistem *database relational* (RDBMS) yang bersifat *open source freeware* di bawah lisensi GPL (*General Public License*). *Database* ini pertama kali dirintis oleh seorang *programmer database* bernama Michael Widenius. Awalnya MySQL hanya berjalan pada sistem Unix dan Linux, namun seiring berjalannya waktu akhirnya diciptakan versi yang dapat digunakan pada hampir semua platform termasuk Windows.

MySQL sebenarnya adalah turunan dari salah satu konsep utama dalam basis data yaitu SQL (*Structured Query Language*). SQL sendiri merupakan bahasa terstruktur yang berfungsi sebagai penghubung antara program *database* dengan bahasa pemrograman. Maksud dari terstruktur adalah adanya beberapa aturan penggunaan yang telah distandarkan dengan standar ANSI (*American National Standards Institute*) atau ISO SQL.

MySQL memiliki fasilitas API (*Application Programming Interface*) yang memungkinkan basis data MySQL mampu diakses oleh berbagai macam bahasa pemrograman. Kemampuan ini merupakan salah satu dari beberapa kelebihan MySQL. Kelebihan MySQL yang lain di antaranya adalah kemampuannya dalam menangani data dalam skala yang besar (lebih dari 50 juta *record*, 60 ribu tabel dan 5 milyar baris), dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah (*multiuser*), mendukung berbagai tipe data yang kompleks dan perlindungan keamanan data yang baik.

2.4 Delphi 7

Delphi 7 merupakan salah satu versi dari program Borland Delphi yang dirilis pada bulan Agustus tahun 2002 oleh Borland *Software* Corporation. Meski telah dikembangkan versi – versi Delphi terbaru, banyak *programmer* yang masih menggunakan Delphi versi ini. Ini disebabkan oleh kemampuan dari Delphi 7 yang masih sangat memadai, mempunyai kestabilan tinggi dan karena versi ini tidak membutuhkan perangkat keras yang terlalu tinggi.

Secara umum, Borland Delphi merupakan sebuah program untuk membuat aplikasi – aplikasi berbasis Windows yang menggunakan bahasa Pascal sebagai bahasa pemrogramannya. Delphi mengusung konsep lingkungan kerja IDE (*Integrated Development Environment*) dan mendukung konsep OOP (*Object Oriented Programming*). Dengan adanya kedua konsep ini, Delphi menjadi *software* pembuat aplikasi yang mudah digunakan. Konsep IDE akan membantu programmer dalam mendesain tampilan secara visual dan menuliskan *coding* antarmuka aplikasi. Sementara konsep OOP akan memudahkan perancangan *software* yang kompleks dengan cara membuat pendekatannya.

Selain dalam pembuatan aplikasi, Delphi juga menyediakan fasilitas yang lengkap dalam pengolahan *database*. Berbagai format *database* yang dapat diolah oleh Delphi di antaranya adalah Paradox, dBase, MS Access, ODBC, SyBASE dan Oracle.

2.5 Microsoft SAPI (*Speech Application Programming Interface*)

Speech Application Programming Interface (SAPI) merupakan sebuah API yang digunakan sebagai pengenal suara di dalam lingkungan pemrograman Windows. Aplikasi yang dikembangkan oleh Microsoft ini telah dikemas baik berupa SDK (*System Development Kit*) maupun disertakan dalam sistem operasi Windows itu sendiri.

BAB III
METODOLOGI PENELITIAN

Penyusunan laporan ini didasarkan pada masalah yang bersifat aplikatif, yakni perancangan dan pembuatan alat agar dapat bekerja sesuai dengan yang direncanakan. Langkah – langkah yang digunakan untuk merealisasikan alat yang dirancang adalah penentuan spesifikasi alat, studi literatur, perancangan dan pembuatan alat, pengujian dan analisis, pengambilan kesimpulan dan penyusunan laporan.

3.1 Perancangan Alat

Perancangan Sistem Antrean dengan Pemanggilan Nama pada Pemungutan Suara Menggunakan RFID meliputi tahap – tahap sebagai berikut :

1) Penentuan spesifikasi alat.
2) Pembuatan diagram blok sistem secara keseluruhan.
3) Penentuan antarmuka untuk menghubungkan seluruh perangkat keras yang digunakan.
4) Perancangan perangkat lunak komputer untuk mengendalikan sistem secara keseluruhan.

3.2 Pembuatan Alat

1) Pembuatan sistem berdasarkan perancangan.
2) Penggabungan tiap – tiap perangkat keras menggunakan antarmuka yang sesuai.
3) Pembuatan perangkat lunak komputer sesuai dengan *flowchart* yang telah dirancang.

3.3 Pengujian Alat

Pengujian alat dilakukan untuk mengetahui kinerja dari sistem yang telah dibuat dan mengetahui tingkat pencapaian dari tujuan yang direncanakan. Pengujian akan dilakukan pada masing – masing bagian dan kemudian dilanjutkan

dengan pengujian sistem secara keseluruhan. Secara gari besar pengujian yang dilakukan adalah sebagai berikut :

1) Pengujian jarak pembacaan tag RFID oleh RFID *Starter Kit*

   Pengujian ini bertujuan untuk menganalisis jarak baca dari RFID *Starter Kit* yang digunakan. Pengujian dilakukan dengan cara mendekatkan beberapa tag RFID yang berbeda pada RFID *Starter Kit* dengan jarak yang bervariasi, yakni mulai dari 1 cm hingga 12 cm. Hasil yang diharapkan adalah jangkauan pembacaan RFID *Starter Kit* untuk membaca informasi pada tag RFID dapat diketahui dengan tepat.

2) Pengujian pencarian data *voter* dalam *database* dan menyimpan data dalam urutan antrean

   Pengujian ini bertujuan untuk menganalisis ketepatan *software* yang dibuat dalam mencari data *voter* yang ada dalam *database* serta menyimpan data *voter* dalam urutan antrean. Pengujian dilakukan dengan menggunakan *software* Delphi 7 yang telah dihubungkan dengan file *database voter*. Dengan memasukkan nomor ID dari *voter*, data *voter* akan ditampilkan oleh *software* yang dibuat dan kemudian dapat dimasukkan ke dalam antrean pada suatu *database* yang baru. Hasil yang diharapkan adalah *software* yang dibuat mampu menampilkan data *voter* sesuai dengan nomor ID yang diinputkan serta mampu menyimpan data *voter* dalam *database* antrean secara berurutan.

3) Pengujian pemanggilan urutan antrean *voter* menurut nomor bilik

   Pengujian ini bertujuan untuk menganalisis ketepatan pemanggilan data *voter* secara audio dan visual sesuai dengan urutan antrean dan nomor bilik. Pengujian dilakukan dengan memasukkan beberapa data dalam suatu *database* urutan antrean yang akan dipanggil melalui *software* yang dibuat menggunakan Delphi 7. Output pemanggilan akan dikeluarkan melalui *speaker* serta monitor. Hasil yang diinginkan adalah data *voter* yang dipanggil akan tepat dan sesuai dengan urutan dan nomor bilik.

4) Pengujian perubahan status pemungutan suara *voter*

   Pengujian bertujuan untuk menganalisis kemampuan sistem dalam menyimpan data *voter* yang telah melakukan pemungutan suara. Pengujian

dilakukan dengan memasukkan beberapa data *voter* yang belum melakukan pemungutan suara dalam suatu *database*. Beberapa data *voter* ini kemudian dipanggil secara visual *software* yang dibuat melalui Delphi 7 dan disimulasikan untuk melakukan pemungutan suara secara sederhana. Untuk mengetahui perubahan status pemungutan suara selanjutnya dapat dilakukan dengan memasukkan nomor ID dari *voter* yang diinginkan melalui menu yang telah disediakan dalam *software* yang dibuat. Hasil yang diharapkan adalah status pemungutan suara *voter* dapat berubah ketika pemungutan suara telah dilakukan.

5) Pengujian sistem secara keseluruhan

Pengujian sistem secara keseluruhan dimulai dari pengidentifikasian tag RFID melalui RFID *Starter Kit*. Hasil pembacaan dari RFID *Starter Kit* akan langsung memanggil data yang ada dalam *database* jika data ditemukan. Kemudian dilanjutkan dengan penyimpanan data dalam tabel antrean dan diakhiri dengan pemanggilan data melalui *speaker* dan monitor. Hasil yang diharapkan adalah sistem mampu membaca informasi yang ada dalam tag RFID, mencari data *voter* dalam *database*, memanggil data *voter* sesuai urutan pembacaan tag RFID dan mampu menyimpan status pemungutan suara *voter* dengan tepat.

BAB IV

PERANCANGAN DAN PEMBUATAN

Dalam perancangan dan pembuatan sistem antrean dengan pemanggilan nama pada pemungutan suara menggunakan RFID ini dilakukan secara bertahap untuk memudahkan dalam analisis setiap bagian sistem maupun sistem secara keseluruhan. Perancangan dan pembuatan sistem ini meliputi perancangan diagram blok sistem secara keseluruhan, perancangan diagram alir (*flowchart*) sistem, pengaturan perangkat keras yang digunakan serta pembuatan *software* sistem. Di samping itu, aspek lain seperti penentuan spesifikasi sistem yang dirancang serta prinsip kerja sistem juga dijelaskan dalam bab ini.

4.1 Penentuan Spesifikasi Alat

Sistem antrean dengan pemanggilan nama yang dirancang memiliki spesifikasi sebagai berikut:

a) Identifikasi *voter* dilakukan dengan menggunakan tag RFID dan RFID *Starter Kit*.
b) Tag RFID yang digunakan adalah tipe tag RFID pasif TK4100.
c) RFID *Starter Kit* yang menggunakan *reader* ID-12 dapat membaca dengan radius kurang lebih 12 cm.
d) RS232 to USB *Converter* digunakan untuk menghubungkan antara RFID *Starter Kit* dengan PC.
e) Kapasitas bilik suara sebanyak tiga buah.
f) *Personal Computer* yang digunakan menggunakan *Operating System* Windows 7 64 bit yang berfungsi sebagai pengolah data, penampil data serta menyimpan informasi dalam *database*.
g) Jumlah antrean *voter*/ jumlah *user* yang disediakan maksimal 999 orang.
h) Pemanggilan antrean akan dilakukan dengan pemanggilan nomor antrean, nama *voter* dan nomor bilik suara melalui *speaker* serta monitor yang disediakan.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

i) Pemanggilan nama dilakukan dengan mengonversi tulisan menjadi suara melalui *software* Borland Delphi 7 menggunakan fungsi *text to speech* dan *database* bahasa Inggris buatan Microsoft (Microsoft SAPI).

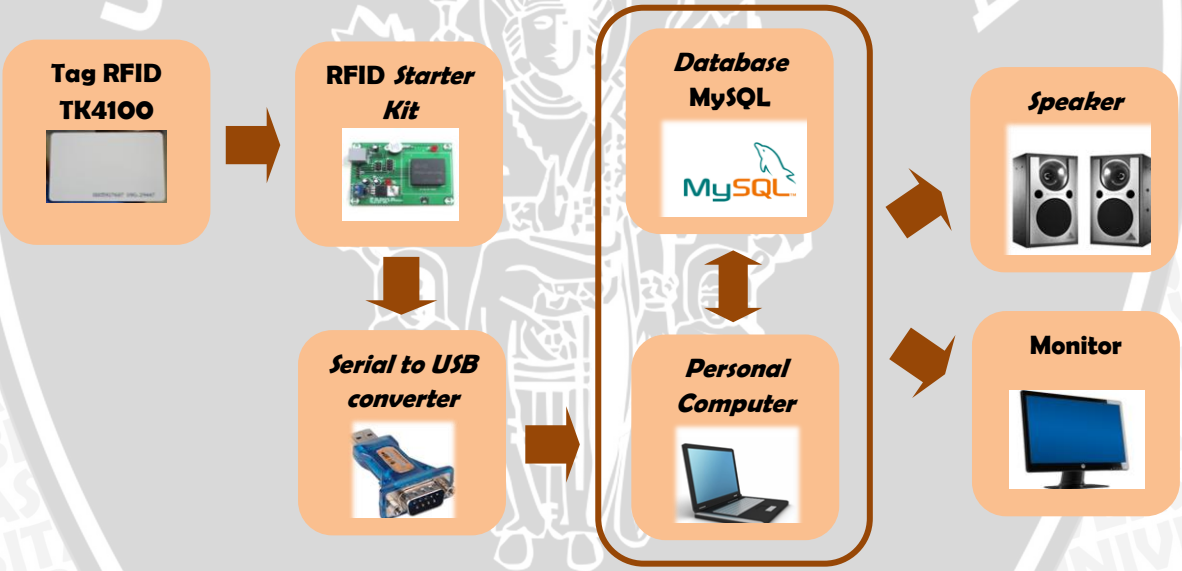
j) Sistem dapat melakukan pemanggilan ulang untuk *voter* yang belum hadir.

k) Program tampilan pada layar monitor dirancang dengan menggunakan *software* Borland Delphi 7.

l) Media penyimpan data (*database*) dibuat menggunakan MySQL.

4.2 Perancangan Alat

Perancangan ini didasarkan pada diagram blok yang ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram blok sistem

Penjelasan mengenai fungsi tiap bagian pada diagram blok dalam Gambar 4.1 adalah sebagai berikut :

a) Tag RFID TK4100 berfungsi sebagai sarana identitas *voter* untuk melakukan pemungutan suara.

b) RFID *Starter Kit* berfungsi untuk membaca nomor identitas dari *voter* yang ada pada tag RFID. RFID *Starter Kit* tersambung dengan PC (*Personal Computer*) operator pemungutan suara.

c) Serial to USB *converter* berfungsi sebagai penghubung RFID *Starter Kit* dengan PC. Bagian ini diperlukan karena PC yang digunakan tidak dapat tersambung langsung dengan keluaran dari RFID *Starter Kit* yang menggunakan port serial.

d) PC berfungsi untuk menjalankan program dari sistem secara keseluruhan, mulai dari pencarian data dalam *database*, menampilkan data pada monitor, pemanggilan *voter* serta penyimpanan status pemungutan suara *voter*.

e) *Database* MySQL berfungsi sebagai tempat untuk meletakkan data *voter* secara keseluruhan. *Database* inilah yang nantinya akan diakses oleh sistem ketika melakukan pencarian data *voter*.

f) *Speaker* berfungsi sebagai media untuk memanggil *voter* yang mendapatkan giliran antrean.

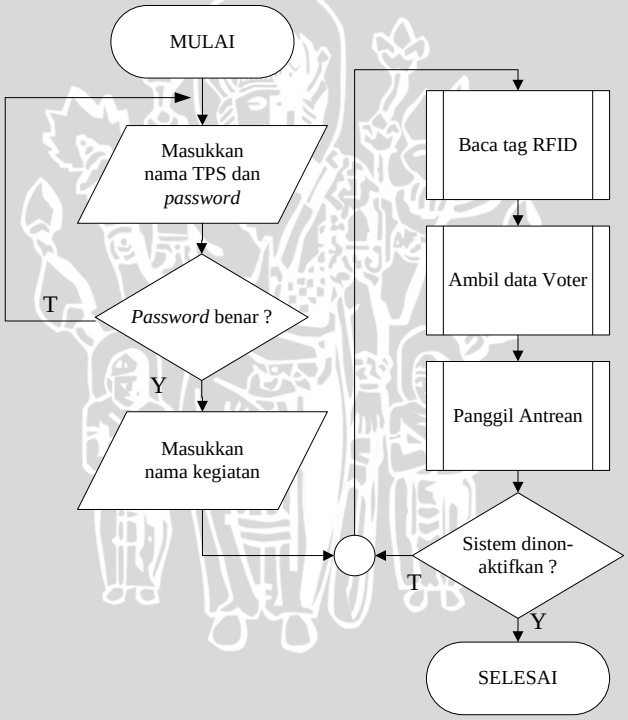
g) Monitor berfungsi sebagai penampil data *voter* yang mendapatkan giliran untuk melakukan pemungutan suara.

Cara kerja sistem antrean ini dimulai dengan identifikasi *voter*. Identifikasi *voter* dilakukan dengan menggunakan *RFID Starter Kit*. Kartu tanda pengenal *voter* yang berupa tag RFID diperlihatkan kepada petugas pemungutan suara untuk kemudian dibaca oleh *RFID Starter Kit*. Hal ini dilakukan untuk mengecek data – data dari *voter*. Pengecekan dilakukan dengan mengakses *database* yang ada pada komputer petugas pemungutan suara. Apabila data *voter* telah ditemukan, data tersebut akan ditampilkan pada monitor petugas dan akan disimpan dalam suatu urutan antrean untuk pelaksanaan pemungutan suara.

Ketika sistem pelaksanaan antrean diaktifkan oleh operator (petugas pemungutan suara), petugas dapat memilih nomor bilik yang akan digunakan *voter* dalam melakukan pemungutan suara. Setelah operator memilih nomor bilik, sistem secara otomatis akan mengambil data urutan antrean yang sebelumnya telah tersimpan. Dari data tersebut kemudian akan dipanggil nomor antrean, nama *voter*, serta nomor bilik suara yang dapat digunakan oleh *voter* sesuai urutan antrean. Output suara merupakan gabungan dari hasil konversi nomor antrean, nama *voter* serta nomor bilik suara menjadi suara melalui *software* Borland Delphi 7. Selain dipanggil melalui speaker, data *voter* yang mendapat giliran

antrean akan ditampilkan pada monitor sehingga dapat menghindari kesalahpahaman apabila terdapat kesamaan nama.

Sesuai dengan cara kerja sistem yang dirancang, tiap – tiap bagian diagram blok ini nantinya akan dihubungkan secara keseluruhan dengan menggunakan antarmuka yang sesuai. Selain itu, dilakukan pula perancangan perangkat lunak yang berguna untuk mengendalikan sistem secara keseluruhan. Perangkat lunak yang berupa program pengontrol sistem ini terdiri atas dua bagian, yakni program tampilan dan *database*. Untuk program tampilan akan dibuat melalui *software* Borland Delphi 7 dengan menggunakan bahasa pemrograman Pascal, sedangkan *database* dibuat melalui *software* MySQL. Keseluruhan rancangan perangkat lunak dari sistem ini tersusun atas beberapa subrutin yang didasarkan pada *flowchart* yang ditunjukkan dalam Gambar 4.2.



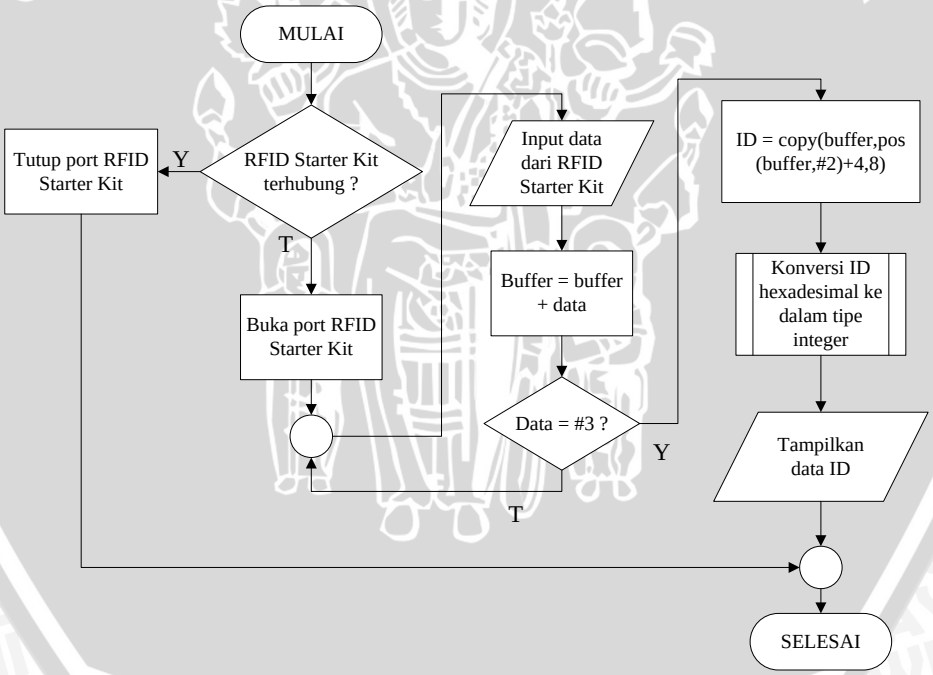
Gambar 4.2 *Flowchart* perancangan perangkat lunak sistem secara keseluruhan

Ketika program sistem dimulai, operator (petugas pemungutan suara) diminta untuk memasukkan nama TPS (Tempat Pemungutan Suara) serta *password* dari sistem itu sendiri. Hal ini bertujuan agar sistem dapat terjaga dari orang – orang yang tidak bertanggungjawab. Setelah *password* yang dimasukkan tepat, program akan meminta operator untuk memasukkan nama kegiatan pemungutan suara yang akan dilakukan. Selanjutnya, program akan memanggil

subrutin pembacaan tag RFID yang berfungsi untuk membaca informasi yang tersimpan dalam tag RFID. Setelah informasi di dalam tag diterima, program memanggil subrutin pengambilan data *voter* untuk mengidentifikasi data – data *voter* yang diperlukan dan menyimpan data tersebut ke dalam urutan antrean. Apabila antrean dimulai, program akan memanggil subrutin Pemanggil Antrean untuk memanggil *voter* sesuai dengan urutan antrean yang telah disimpan.

4.2.1 Perancangan Subrutin Pembacaan Tag RFID oleh RFID *Starter Kit*

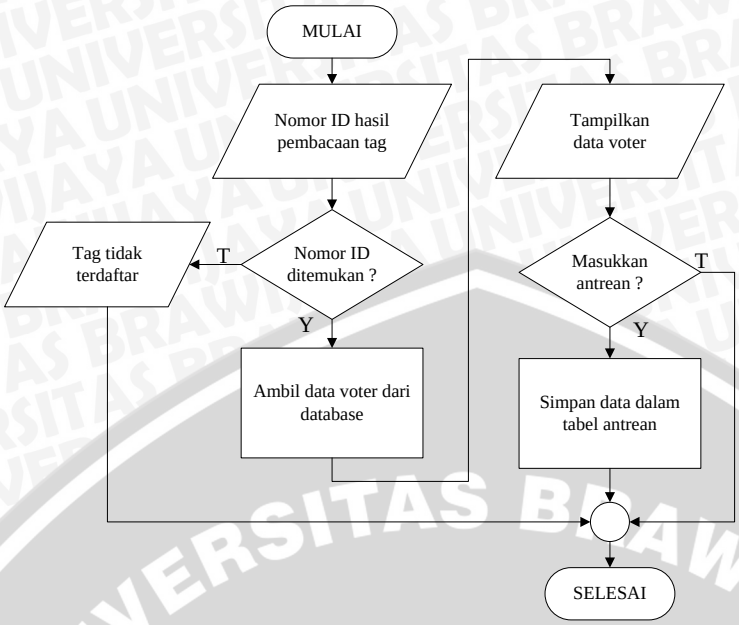
Perancangan ini bertujuan untuk membaca informasi yang ada dalam tag RFID dan menampilkannya pada program penampil. Dalam perancangan ini, port awalnya dibuka dan kemudian mulai untuk membaca data yang ada pada tag RFID. Setelah data diterima oleh RFID *Starter Kit* dan kemudian dikirimkan menuju PC, data akan dikonversi menjadi suatu nilai dengan tipe data *integer* yang nantinya dapat dicocokkan dengan nomor ID yang tercetak pada tag RFID. *Flowchart* dari subrutin ini dapat ditunjukkan dalam Gambar 4.3.



Gambar 4.3 *Flowchart* perancangan subrutin pembacaan tag RFID oleh RFID *Starter Kit*

4.2.2 Perancangan Subrutin Pencarian Data *Voter* dalam *Database*

Perancangan ini bertujuan untuk melakukan pencarian data dalam *database voter* yang telah dibuat sebelumnya dan menyimpan data tersebut dalam suatu urutan antrean. *Flowchart* dari subrutin ini ditunjukkan dalam Gambar 4.4.



Gambar 4.4 *Flowchart* perancangan subrutin pencarian data *voter* dalam *database*

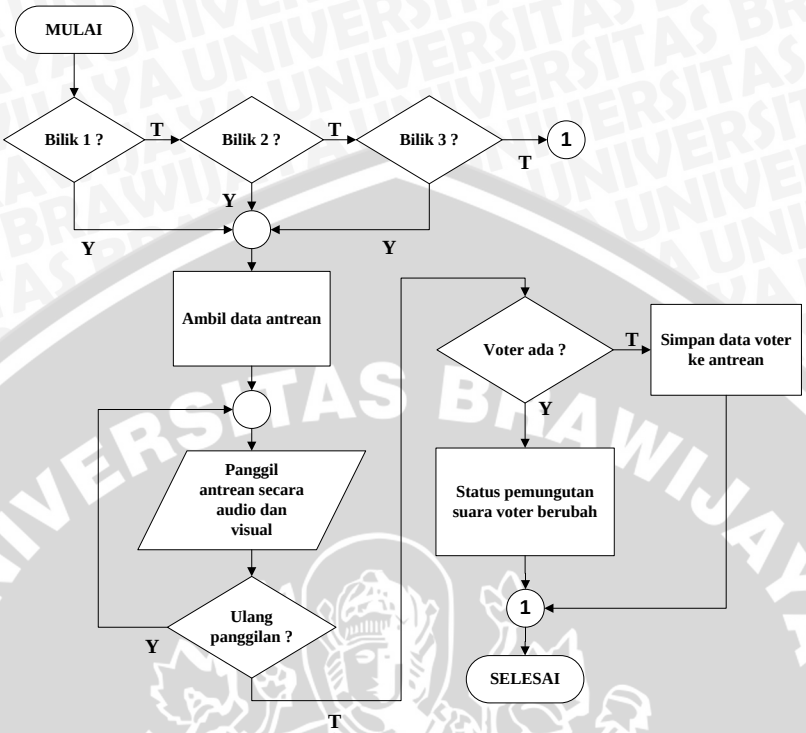
Nomor ID yang merupakan informasi yang telah diterima oleh sistem pada subrutin pembacaan tag oleh RFID *Starter Kit* akan menjadi kunci untuk pencarian data *voter* di dalam *database*. Hal ini disebabkan oleh nomor tag yang selalu berbeda untuk tiap – tiap *voter* sehingga tidak akan terjadi adanya data ganda untuk sebuah nomor ID. Ketika nomor ID yang dicari ditemukan di dalam *database*, maka data dari *voter* dengan nomor ID tersebut akan ditampilkan pada program tampilan. Selanjutnya, *voter* dapat dimasukkan ke dalam antrean apabila status pemungutan *voter* menyatakan bahwa *voter* tersebut belum melakukan pemungutan suara. Hal ini bertujuan untuk menghindari adanya pemungutan suara ganda untuk satu identitas *voter*.

Untuk suatu keadaan khusus, dimana tag RFID *voter* ternyata hilang, maka disediakan suatu pilihan pencarian data *voter* menggunakan identitas nama dari *voter* tersebut. *Flowchart* untuk keadaan ini hampir sama dengan *flowchart* dalam Gambar 4.4, namun nomor ID yang digunakan sebagai kunci pencarian data di dalam *database* diganti dengan nama lengkap dari *voter* yang bersangkutan.

4.2.3 Perancangan Subrutin Pemanggilan Antrean *Voter*

Perancangan ini bertujuan untuk melakukan pemanggilan kepada *voter* yang mendapatkan giliran antrean untuk melakukan pemungutan suara. Pemanggilan akan dilakukan sesuai dengan nomor bilik yang dipilih oleh operator

dengan urutan panggilan berupa nomor urut antrean, nama *voter* serta nomor bilik yang digunakan. *Flowchart* dari subrutin ini ditunjukkan dalam Gambar 4.5.



Gambar 4.5 *Flowchart* perancangan subrutin pemanggilan antrean *voter*

Ketika operator melakukan pemanggilan pada nomor bilik tertentu, sistem akan mencari *voter* dengan nomor antrean terendah yang belum melakukan pemungutan suara. Data berupa nomor antrean, nama *voter* serta foto akan ditampilkan secara visual pada monitor sedangkan data berupa audio akan memanggil nomor antrean, nama *voter* serta nomor bilik. Untuk menjaga agar *voter* yang bersangkutan tidak dipanggil lagi pada bilik lain pada saat bersamaan, status pemungutan suara *voter* sementara diubah menjadi telah melakukan pemungutan suara. Ketika *voter* memenuhi panggilan antrean, maka *voter* dianggap telah melakukan pemungutan suara dan status pemungutan suaranya akan tetap tersimpan. Sebaliknya,apabila ternyata *voter* tidak memenuhi panggilan, maka urutan antrean *voter* tersebut akan diubah dan diletakkan pada urutan terbawah antrean yang telah ada.

4.2.4 Perancangan *Database*

Perancangan ini berfungsi untuk membuat suatu tempat penyimpanan segala macam data yang diperlukan dalam sistem antrean pemungutan suara ini.

Dalam sistem ini dirancang sebuah *database* yang terdiri atas dua buah tabel, yakni tabel utama dan tabel antrean.

Tabel utama, yakni tabel biodata *voter* berisi data – data berupa nomor ID, nama, jenis kelamin, alamat, foto, status pemilihan, nama TPS, nomor bilik serta nomor antrean *voter*. Atribut dan tipe data masing –masing kolom tabel utama ditunjukkan dalam Tabel 4.1.

Tabel 4.1 Atribut – Atribut Tabel Utama *Database*

No	Nama Tabel	Nama Kolom	Tipe Data	*Field Size*	Keterangan
1	Tabel Utama	ID	*Integer*	10	*Primary key*
2		Nama	*Varchar*	50	
3		Jenis Kelamin	*Tinyint*	1	
4		Alamat	*Varchar*	100	
5		Foto	*Varchar*	100	
6		Status	*Tinyint*	1	
7		Bilik	*Integer*	1	
8		No. Antrean	*Integer*	2	
9		TPS	*Varchar*	20	

ID merupakan nomor identitas dari setiap *voter* yang tersimpan dalam tag. Karena nomor dari tag ini selalu berbeda satu sama lain, nomor inilah yang dijadikan kunci (*primary key*) dalam pencarian data *voter*. Data berupa nama, jenis kelamin, alamat serta foto merupakan identitas dari masing – masing *voter* yang sebelumnya telah diisikan secara manual. Untuk data status, nama TPS, bilik serta nomor antrean akan terisi secara otomatis ketika antrean dilakukan.

Tabel yang kedua, yakni tabel antrean berisi informasi nomor antrean *voter*, nomor ID *voter*, nama *voter*, nomor bilik serta status pemilihan *voter*. Atribut dan tipe data masing – masing kolom ditunjukkan dalam Tabel 4.2.

Tabel 4.2 Atribut – Atribut Tabel Antrean *Database*

No	Nama Tabel	Nama Kolom	Tipe Data	*Field Size*	Keterangan
1	Tabel Antrean	No. Antrean	*Integer*	2	*Primary key*
2		ID	*Integer*	10	

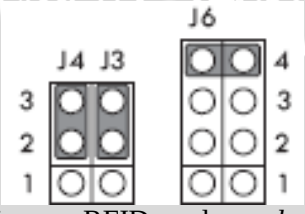
3	Nama	*Varchar*	50
4	Bilik	*Integer*	1
5	Status	*Tinyint*	1

Dalam tabel antrean ini yang menjadi *primary key* adalah nomor antrean dari masing – masing *voter*. Selain karena nomor tersebut tidak akan sama untuk setiap *voter*, yang paling utama adalah adanya kebutuhan untuk pemanggilan *voter* secara berurutan sesuai dengan nomor antrean. Ketika *voter* dengan nomor urutan tertentu telah melakukan pemungutan suara, maka data status akan berubah. Setelah perubahan tersebut maka sistem akan menuju *voter* dengan urutan berikutnya.

4.3 Pembuatan Alat

4.3.1 Pengaturan RFID *Starter Kit*

RFID *Starter Kit* yang digunakan memiliki fasilitas untuk pengaturan mode kerja dari RFID *reader* itu sendiri dan pengaturan format data output yang diinginkan. Dalam hal ini yang diatur adalah *setting jumper* J3 dan J4 yang ada pada RFID *Starter Kit*. Karena yang dibutuhkan dalam tugas akhir ini adalah data dengan format UART RS-232 (ASCII), maka *setting jumper* yang digunakan adalah seperti dalam Gambar 4.6. Selain karena alasan tersebut, pemilihan *setting jumper* seperti dalam Gambar 4.6 juga disebabkan karena dalam sistem yang dibuat hanya memerlukan mode pembacaan saja (*reader only*).



Gambar 4.6 *Setting jumper* RFID mode *reader only* dengan format data UART RS-232 (ASCII)
Sumber : Innovative Electronics, 2007.

4.3.2 Pembuatan *Software*

Sesuai dengan perancangan yang telah dilakukan bahwa *software* yang diperlukan dalam sistem ini terdiri atas dua bagian, yakni *software* berupa program tampilan dan berupa *database*.

a) Program Tampilan pada PC

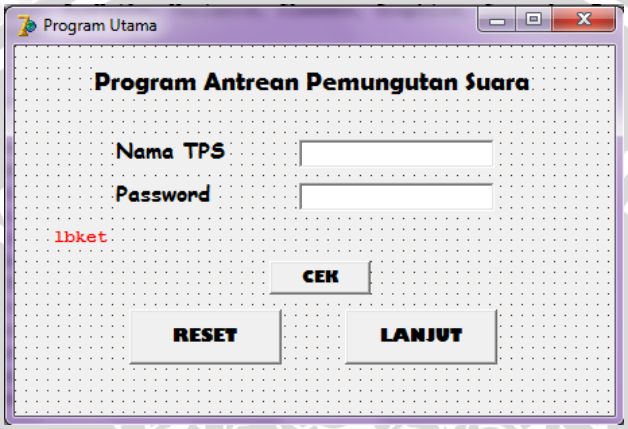
Program tampilan yang dibuat meliputi program tampilan utama, program pengisian nama kegiatan, program identifikasi *voter* dan program pengontrol pemanggilan. Diagram blok tampilan program dapat ditunjukkan dalam Gambar 4.7.

Gambar 4.7 Diagram Blok Program Tampilan pada PC

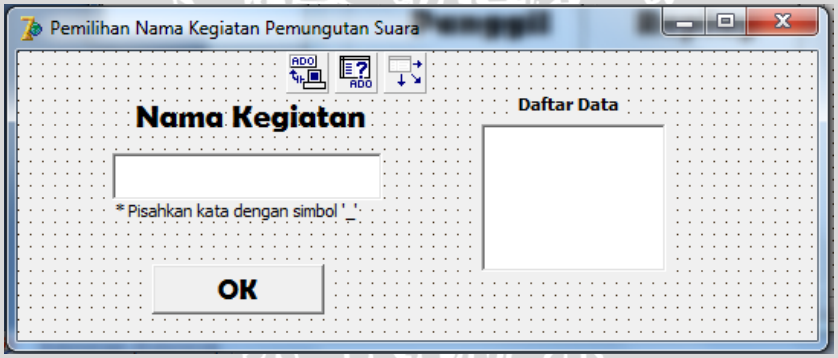
Program tampilan utama berfungsi sebagai pintu masuk seorang operator dalam menjalankan fungsi kontrol antrean. Di dalam program ini akan ditampilkan nama *software*, isian untuk memasukkan nama TPS, isian untuk memasukkan *password* sistem, tombol 'CEK', tombol 'RESET' serta tombol 'LANJUT'. Tombol 'CEK' berfungsi untuk mengetahui ketepatan *password* yang diisikan oleh operator. Ketika *password* yang dimasukkan tepat, maka kedua tombol yang tersedia akan aktif. Tombol 'RESET' berfungsi untuk memulai antrean dari awal lagi. Hal ini berarti seluruh data pemilihan yang sebelumnya ada akan dihapus dan akan dibuat sebuah urutan antrean baru. Tombol 'LANJUT' berfungsi untuk melanjutkan proses antrean yang mungkin sebelumnya dinon-aktifkan untuk istirahat. Hasil pembuatan program tampilan utama ini ditunjukkan dalam Gambar 4.8.

Program Pengisian Nama Kegiatan akan muncul setelah operator menekan baik tombol 'RESET' maupun tombol 'LANJUT'. Seperti ditunjukkan dalam Gambar 4.9, dalam tampilan program ini terdapat

sebuah kotak isian yang berfungsi untuk tempat pengisian nama kegiatan pemungutan suara yang dilakukan. Apabila pada program sebelumnya operator menekan tombol 'RESET', maka operator diharuskan untuk mengisikan nama kegiatan baru tersebut. Apabila ternyata antrean yang akan dilakukan adalah lanjutan dari sebelumnya, maka operator harus mengisikan nama kegiatan yang pilihannya dapat dilihat pada listbox daftar data yang ada di samping kotak pengisian.



Gambar 4.8 Program Tampilan Utama

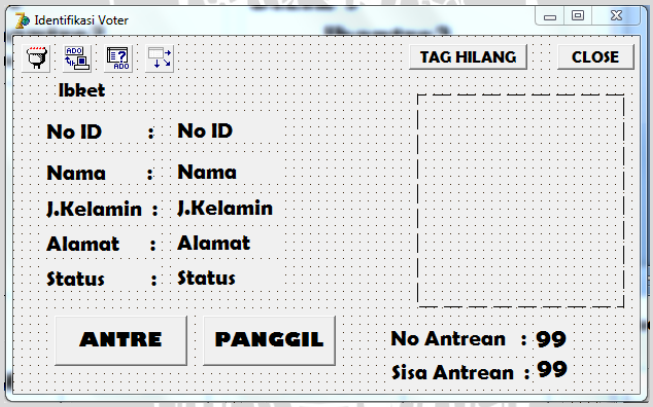


Gambar 4.9 Tampilan Program Pengisian Nama Kegiatan

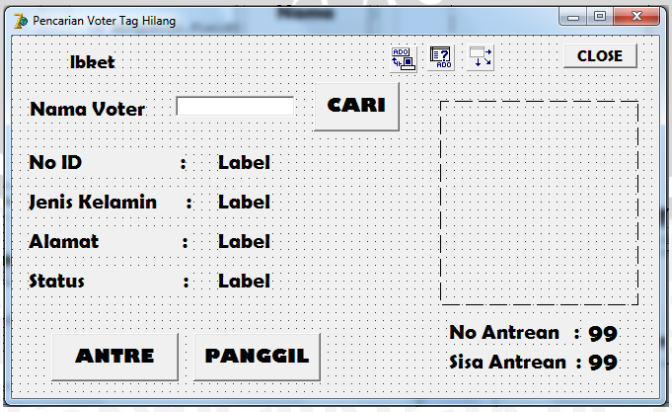
Selanjutnya, setelah operator mengisikan nama kegiatan dan menekan tombol 'OK' pada program pengisian nama kegiatan, program identifikasi *voter* akan muncul. Dalam program ini digunakan beberapa jenis *component palette*, yakni komponen XComDrv yang berfungsi untuk komunikasi dengan RFID *Starter Kit*, komponen ADO yang berfungsi untuk menghubungkan dengan *database* serta komponen *Data Access*.

Sebelum tag didekatkan pada RFID *Starter Kit* dan data belum terbaca, tampilan program akan terlihat seperti dalam Gambar 4.10. Label

Nomor ID, Nama, J. Kelamin, Alamat dan Status masing – masing akan menampilkan informasi berupa nomor ID kartu, nama, jenis kelamin, alamat serta status pemungutan suara dari *voter*. Kotak dengan garis putus – putus merupakan tempat untuk menampilkan foto dari *voter*. Tombol 'PANGGIL', 'TAG HILANG' dan tombol 'CLOSE' akan aktif mulai dari awal munculnya program, sedangkan tombol 'ANTRE' akan aktif ketika sebuah data *voter* ditemukan dan ditampilkan dalam program tersebut. Tombol 'ANTRE' berfungsi untuk menyimpan data *voter* ke dalam antrean pemungutan suara, tombol 'TAG HILANG' berfungsi untuk pencarian data *voter* apabila kartu tag hilang, tombol 'CLOSE' untuk menutup aplikasi secara keseluruhan, sedangkan tombol 'PANGGIL' berfungsi untuk memanggil *voter* yang mendapatkan giliran antrean untuk melakukan pemungutan suara. Tampilan program untuk pencarian data voter yang kehilangan kartu tag RFID ditunjukkan dalam Gambar 4.11.

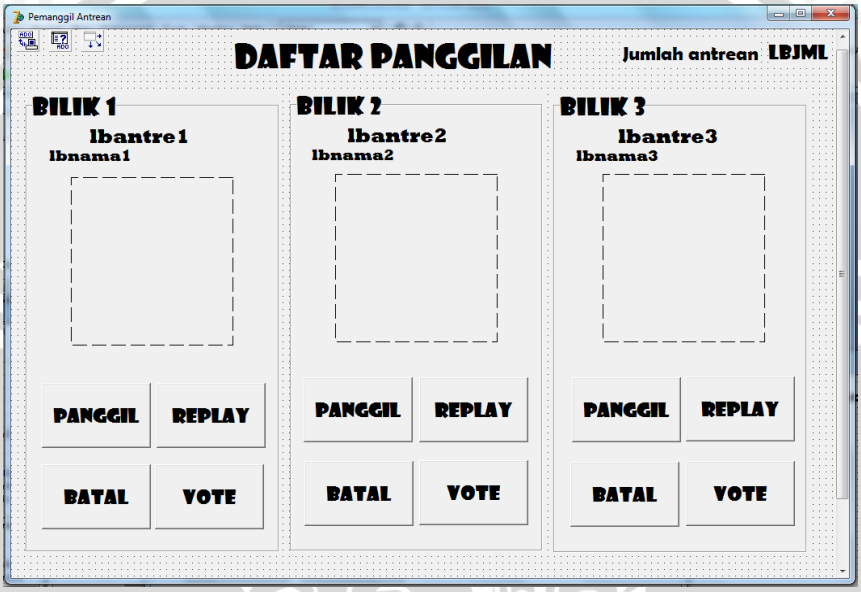


Gambar 4.10 Tampilan Program Identifikasi *Voter*



Gambar 4.11 Tampilan Program Identifikasi *Voter* yang kehilangan tag RFID

Selanjutnya, setelah tombol ‘PANGGIL’ ditekan maka akan muncul program pengontrol panggilan antrean *voter*. Dalam program ini akan ditampilkan data dari *voter* yang mendapatkan giliran antrean sesuai dengan nomor bilik yang digunakan. Data *voter* yang ditampilkan hampir sama dengan data yang ditampilkan dalam program identifikasi *voter*, namun data alamat dan status digantikan dengan nomor urut antrean dari *voter*. Rancangan tampilan dari program ini ditunjukkan dalam Gambar 4.12.



Gambar 4.12 Tampilan Program Pengontrol Pemanggilan *Voter*

Tombol ‘Panggil’ yang ada pada masing – masing bilik berfungsi untuk memanggil *voter* yang mendapat giliran antrean untuk menempati bilik tersebut. Apabila setelah pemanggilan awal *voter* masih belum hadir, tombol ‘Replay’ akan aktif dan dapat digunakan beberapa kali jika ingin melakukan pemanggilan ulang *voter*. Ketika *voter* yang dipanggil tidak kunjung datang, operator dapat menekan tombol ‘Batal’ yang akan mengganti *voter* tersebut dengan *voter* yang ada pada urutan berikutnya. *Voter* yang digantikan tidak akan dihapuskan dari urutan antrean, namun akan diletakkan pada urutan terakhir dari jumlah antrean yang ada. Tombol ‘Vote’ yang disediakan dalam tampilan program tersebut ditujukan untuk perumpaan tombol yang digunakan *voter* dalam

melakukan pemungutan suara. Jadi, ketika tombol tersebut ditekan, *voter* dianggap telah melakukan pemungutan suara.

Pada diagram blok program tampilan pada PC digambarkan bahwa antara blok untuk program tampilan identifikasi *voter* dan program pengontrol pemanggilan *voter* memiliki hubungan dua arah. Maksud dari hal ini adalah bahwa ketika sistem dijalankan, kedua program tersebut dapat dilakukan secara bersamaan. Jadi proses pengidentifikasian *voter* dapat dilakukan di sela – sela pemanggilan antrean *voter*.

b) *Database* Sistem

Seperti yang telah dijelaskan sebelumnya, dalam sistem ini dibuat *database* yang memuat dua buah tabel. Tabel yang pertama berisi seluruh data *voter* yang telah memiliki hak untuk melakukan pemungutan suara dan tabel yang kedua berisi urutan antrean *voter*. *Database* ini dibuat melalui *software* MySQL. Hasil pembuatan *database* sistem ditunjukkan dalam Gambar 4.13(a) dan (b).

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(10)			No	0		Change Drop Browse distinct values Primary
2	nama	varchar(50)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary
3	jenis_kelamin	tinyint(1)			No	None		Change Drop Browse distinct values Primary
4	alamat	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary
5	foto	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary
6	status	tinyint(1)			No	None		Change Drop Browse distinct values Primary
7	bilik	int(1)			No	None		Change Drop Browse distinct values Primary
8	nomor_antrean	int(2)			No	None		Change Drop Browse distinct values Primary
9	tps	varchar(20)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary

(a)

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	no	int(2)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary
2	id	int(15)			Yes	NULL		Change Drop Browse distinct values Primary
3	nama	varchar(50)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary
4	bilik	int(1)			Yes	NULL		Change Drop Browse distinct values Primary
5	status	tinyint(1)			Yes	NULL		Change Drop Browse distinct values Primary

(b)

Gambar 4.13 (a) Hasil pembuatan tabel utama data *voter* pada *database*
(b) Hasil pembuatan tabel antrean *voter* pada *database*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB V
PENGUJIAN DAN ANALISIS

Pengujian dilakukan untuk menganalisis hasil pembuatan alat yang dirancang dalam mencapai tujuan yang direncanakan. Pengujian ini berguna untuk mencari masalah yang mungkin timbul, mengevaluasi jalannya sistem, dan mengadakan perbaikan jika ada kekurangan. Pengujian akan dilakukan secara per blok terlebih dahulu yang kemudian dilanjutkan dengan pengujian secara keseluruhan sistem.

5.1 Pengujian Jarak Pembacaan Tag RFID oleh RFID *Starter Kit*

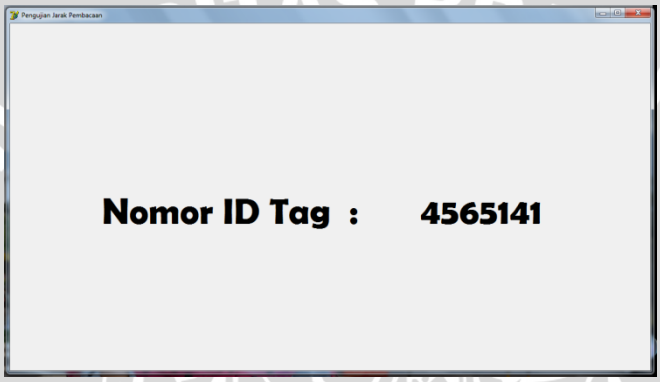
Pengujian ini bertujuan untuk menganalisis jarak baca dari RFID *Starter Kit* dalam membaca informasi yang tersimpan dalam tag RFID. Pengujian dilakukan dengan menggunakan suatu program yang dibuat melalui *software* Delphi 7. Di dalam program tersebut akan ada proses pembacaan data informasi nomor ID tag. *Flowchart* dari pengujian ini sama seperti *flowchart* pembacaan tag RFID yang ditunjukkan dalam Gambar 4.3 sebelumnya.

Dalam pengujian ini, lima tag RFID yang berbeda akan didekatkan pada RFID *Starter Kit* dengan jarak yang bervariasi, yakni mulai dari 1 cm hingga 12 cm secara bergantian. Apabila pada jarak tersebut RFID *Starter Kit* dapat menerima informasi dari tag RFID, maka informasi tersebut akan ditampilkan pada tampilan program. Informasi yang ditampilkan oleh program kemudian akan dibandingkan dengan nomor ID yang tercetak pada masing – masing tag RFID untuk mengetahui ketepatan data yang dibaca oleh RFID *Starter Kit*.

Pengujian yang pertama menggunakan tag dengan nomor ID tercetak 0004565141 seperti dalam Gambar 5.1. Hasil tampilan pengujian yang pertama untuk tag ini pada jarak baca 1 cm ditunjukkan dalam Gambar 5.2. Dalam tampilan tersebut terlihat bahwa nomor ID dari tag telah berhasil terbaca dengan tepat, namun tanpa tiga angka nol di depan. Hal ini disebabkan karena nomor ID yang ditampilkan oleh program merupakan hasil pengubahan data tag dari

heksadesimal menjadi desimal. Untuk hasil pengujian keseluruhan ditunjukkan dalam Tabel 5.1.

Gambar 5.1 Tag RFID yang digunakan untuk pengujian pertama



Gambar 5.2 Tampilan hasil pengujian pertama

Tabel 5.1 Hasil Pengujian Jarak Pembacaan Tag RFID oleh RFID *Starter Kit*

Jarak (cm)	4565141	4565150	5931108	4374549	5927687
1	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
2	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
3	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
4	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
4,5	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
5	Berhasil	Berhasil	Gagal	Berhasil	Gagal
5,5	Gagal	Berhasil	Gagal	Gagal	Gagal
6	Gagal	Berhasil	Gagal	Gagal	Gagal
6,5	Gagal	Gagal	Gagal	Gagal	Gagal
7	Gagal	Gagal	Gagal	Gagal	Gagal
8	Gagal	Berhasil	Gagal	Gagal	Gagal
10	Gagal	Gagal	Gagal	Gagal	Gagal
12	Gagal	Gagal	Gagal	Gagal	Gagal

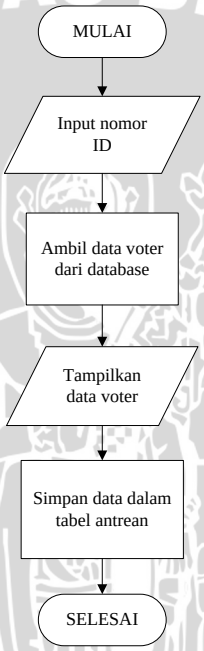
Catatan : Keterangan ‘berhasil’ berarti nomor ID tag yang terbaca sesuai dengan nomor ID yang tercetak pada tag. ‘Gagal’ berarti nomor ID tidak sesuai dengan nomor yang tercetak atau tidak dapat terbaca sama sekali.

Dari hasil keseluruhan pengujian jarak pembacaan tag RFID oleh RFID *Starter Kit* ini terlihat bahwa pada jarak pengujian 5 cm mulai ada nomor ID tag

yang tidak dapat terbaca. Ini berarti bahwa jarak baca maksimal dari RFID *Starter Kit* yang digunakan adalah 4,5 cm.

5.2 Pengujian Pencarian Data *Voter* dalam *Database* dan Menyimpan Data dalam Urutan Antrean

Pengujian ini bertujuan untuk menganalisis ketepatan *software* yang dibuat dalam mencari data *voter* yang ada dalam *database* serta menyimpan data *voter* dalam urutan antrean. Pengujian dilakukan dengan menggunakan *software* Delphi 7 yang telah dihubungkan dengan file *database voter*. *Flowchart* pengujian ini ditunjukkan dalam Gambar 5.3.



Gambar 5.3 *Flowchart* pengujian pencarian data dan penyimpanan data dalam antrean

Dalam pengujian ini, dimasukkan 10 nomor ID dari *voter* yang datanya telah tersimpan di dalam tabel *database* utama. Masing – masing data *voter* dengan nomor ID yang diinputkan akan ditampilkan pada program tampilan yang dibuat. Tampilan data inilah yang akan dibandingkan dengan data *voter* yang ada dalam tabel *database* untuk mengetahui ketepatan pencarian data. Tabel *database voter* dapat dilihat dalam Gambar 5.4. Pengujian akan dilanjutkan dengan memasukkan data – data *voter* yang telah ditampilkan ke dalam urutan antrean pada tabel *database* antrean. Setelah kesepuluh data *voter* dientrikan, akan dilakukan pengecekan pada tabel antrean. Fungsi pengecekan ini adalah untuk

mengetahui kesesuaian urutan penyimpanan data dengan urutan antrean yang dihasilkan.

Hasil pengujian yang pertama, yakni dengan input nomor ID 4374549, hasil tampilan program diperlihatkan dalam Gambar 5.5. Setelah dibandingkan dengan data yang ada dalam tabel *database voter*, pengujian pencarian data *voter* yang pertama dinyatakan berhasil. Selanjutnya dilakukan pengecekan pada tabel *database* antrean seperti ditunjukkan dalam Gambar 5.6. Hasil dari pengecekan ini terlihat bahwa *voter* dengan nomor ID 4374549 telah berhasil tersimpan dalam antrean dengan nomor antrean 1. Hasil pengujian pencarian data *voter* dan penyimpanan data dalam antrean ditunjukkan dalam Tabel 5.2.

id	nama	jenis_kelamin	alamat	foto	status	bilik	nomor_antrean	tps
4374549	Pierre Tendean	1	Jl. Jakarta 1	E:\foto\baru\16pierre tendean.bmp	0	0	0	
4374555	Ahmad Yani	1	Jl.Ananas 25	E:\foto\baru\3Ahmad_Yani.bmp	0	0	0	
4374560	Cut Nyak Dhien	0	Jl.Veteran A2	E:\foto\baru\5cut nyak dhien.bmp	0	0	0	
4565141	Adam Malik	1	Jl.D.Kerinci 3	E:\foto\baru\1adam malik.bmp	0	0	0	
4565150	Dewi Sartika	0	Jl.Bukittinggi 5D	E:\foto\baru\7dewi sartika.bmp	0	0	0	
4565151	Soekarno	1	Jl.Bumi Indah 8	E:\foto\baru\9irsoekarno.bmp	0	0	0	
4565156	Nyi Ageng Serang	0	Jl.Nusa Indah 9	E:\foto\baru\13nyi ageng serang.bmp	0	0	0	
5927687	Martha Christina	0	Jl.Ranu Pani 67	E:\foto\baru\11martha christina tiahahu.bmp	0	0	0	
5927691	Cut Nyak Meutia	1	Jl.Sumbersari 3	E:\foto\baru\6cut nyak meutia.bmp	0	0	0	
5931085	Soedirman	1	Jl.Kedondong B1	E:\foto\baru\10jenderal sudirman.bmp	0	0	0	
5931108	Mohammad Hatta	1	Jl.Kawi 1C	E:\foto\baru\12moh hatta.bmp	0	0	0	
5931891	Kartini	0	Jl.Bandung 33	E:\foto\baru\17ra kartini.bmp	0	0	0	

Gambar 5.4 Tabel utama *database voter*

Gambar 5.5 Hasil pengujian pertama pencarian data *voter*

no	id	nama	bilik	status
1	4374549	Pierre Tendean	0	0

Gambar 5.6 Hasil pengecekan tabel *database* antrean untuk pengujian pertama

Tabel 5.2 Hasil Pengujian Pencarian Data *Voter* dalam *Database*

Tag	Input Nomor ID	Nama	JK	Alamat	Status	Ket.
1	4374549	Pierre	L	Jl. Jakarta 1	Belum	Berhasil

		Tendean			memilih	
2	5931891	Kartini	P	Jl. Bandung 33	Belum memilih	Berhasil
3	4374560	Cut Nyak Dhien	P	Jl. Veteran A2	Belum memilih	Berhasil
4	4565141	Adam Malik	L	Jl. D. Kerinci 3	Belum memilih	Berhasil
5	4374555	Ahmad Yani	L	Jl. Ananas 25	Belum memilih	Berhasil
6	4565150	Dewi Sartika	P	Jl. Bukittinggi 5D	Belum memilih	Berhasil
7	5931085	Soedirman	L	Jl. Kedondong B1	Belum memilih	Berhasil
8	5927691	Cut Nyak Meutia	P	Jl. Sumbersari 3	Belum memilih	Berhasil
9	4565151	Soekarno	L	Jl. Bumi Indah 8	Belum memilih	Berhasil
10	4565156	Nyi Ageng Serang	P	Jl. Nusa Indah 9	Belum memilih	Berhasil

Catatan : JK : Jenis Kelamin ; L : Laki – Laki ; P : Perempuan ; Ket : Keterangan

Dari tabel hasil pengujian pencarian data *voter* secara keseluruhan terlihat bahwa program telah mampu menemukan data *voter* di dalam *database* secara tepat. Setelah dilakukan pengecekan pada tabel *database* antrean, urutan antrean pada tabel tersebut telah sesuai dengan urutan memasukkan nomor ID. Tabel *database* antrean ditunjukkan dalam Gambar 5.7. Dari kedua rangkaian pengujian ini disimpulkan bahwa program telah mampu mencari data *voter* pada tabel *database* antrean dengan tepat dan mampu menyimpan urutan antrean *voter*.

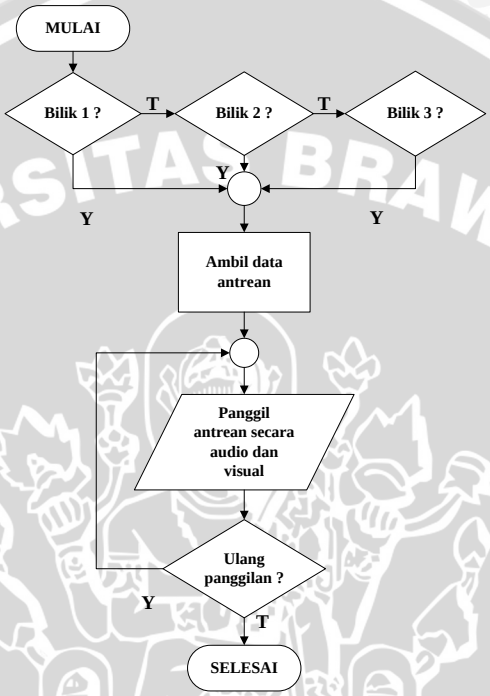
no	id	nama	bilik	status
1	4374549	Pierre Tendean	0	0
2	5931891	Kartini	0	0
3	4374560	Cut Nyak Dhien	0	0
4	4565141	Adam Malik	0	0
5	4374555	Ahmad Yani	0	0
6	4565150	Dewi Sartika	0	0
7	5931085	Soedirman	0	0
8	5927691	Cut Nyak Meutia	0	0
9	4565151	Soekarno	0	0
10	4565156	Nyi Ageng Serang	0	0

Gambar 5.7 Tabel *database* antrean hasil penyimpanan urutan antrean *voter*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

5.3 Pengujian Pemanggilan Urutan Antrean *Voter* Menurut Nomor Bilik

Pengujian ini bertujuan untuk menganalisis ketepatan pemanggilan data *voter* secara audio dan visual sesuai dengan urutan antrean dan nomor bilik. Pengujian dilakukan dengan menggunakan program yang dibuat melalui *software* Delphi 7 yang telah dihubungkan dengan *database* MySQL. *Flowchart* pengujian ini ditunjukkan dalam Gambar 5.8.



Gambar 5.8 *Flowchart* pengujian pemanggilan urutan antrean *voter* menurut nomor bilik

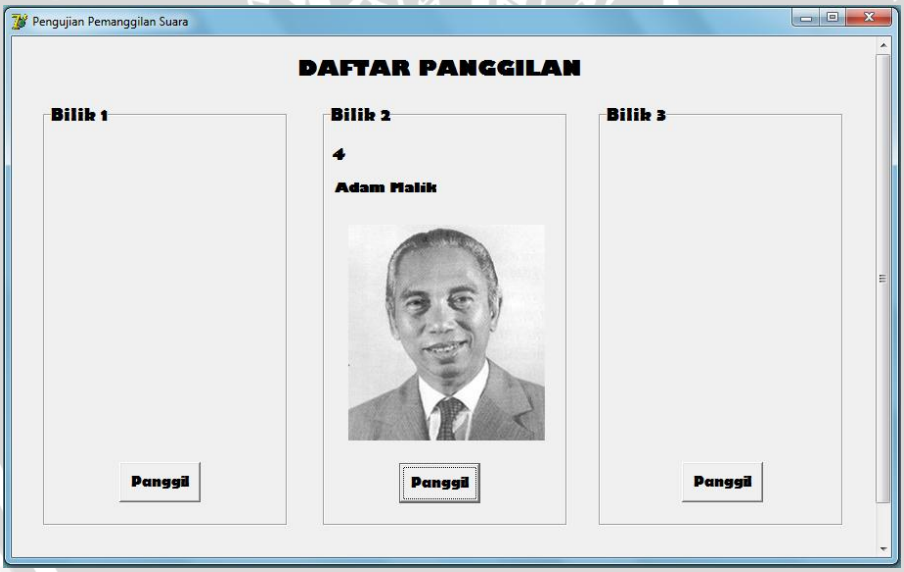
Sebelum melakukan pengujian, di dalam tabel *database* antrean akan diberikan 10 urutan antrean dengan nomor yang acak seperti ditunjukkan dalam Gambar 5.9. Pengacakan nomor ini ditujukan untuk mengetahui kemampuan program dalam memanggil antrean secara berurutan. Pada saat program pengujian dijalankan, pemanggilan dilakukan dengan cara pemilihan bilik acak. Hasil pemanggilan *voter*, baik audio maupun visual kemudian dicatat dan dibandingkan dengan data *voter* yang ada di dalam tabel *database* utama yang ditunjukkan dalam Gambar 5.4 sebelumnya. Selain itu, urutan pemanggilan *voter* juga akan dibandingkan dengan data antrean yang ada di dalam tabel *database* antrean.

Pada pengujian yang pertama dilakukan pemanggilan *voter* pada bilik 2. Hasil pengujian secara visual ditunjukkan dalam Gambar 5.10. Dalam tampilan tersebut tertulis bahwa *voter* yang dipanggil menempati bilik 2 mempunyai nomor

antrean 4 dengan nama Adam Malik. Secara audio, pengujian yang pertama ini menghasilkan output suara yang berbunyi "*Nomor empat Adam Malik menuju bilik 2*". Setelah dilakukan pengecekan, kedua bentuk output pengujian pertama ini telah sesuai dengan data pada tabel *database* antrean dan tabel *database voter*. Hasil pengujian pemanggilan *voter* secara keseluruhan ditampilkan dalam Tabel 5.3.

no	id	nama	bilik	status
4	4565141	Adam Malik	0	0
10	4565151	Soekarno	0	0
11	4565156	Nyi Ageng Serang	0	0
70	5931891	Kartini	0	0
85	4374555	Ahmad Yani	0	0
99	4565150	Dewi Sartika	0	0
100	5927687	Martha Christina	0	0
191	5927691	Cut Nyak Meutia	0	0
333	5931108	Mohammad Hatta	0	0
851	5931085	Soedirman	0	0

Gambar 5.9 Tabel *database* antrean yang digunakan sebagai acuan pemanggilan



Gambar 5.10 Tampilan visual hasil pengujian pemanggilan *voter* yang pertama

Tabel 5.3 Hasil Pengujian Pemanggilan Urutan Antrean *Voter* Menurut Nomor Bilik

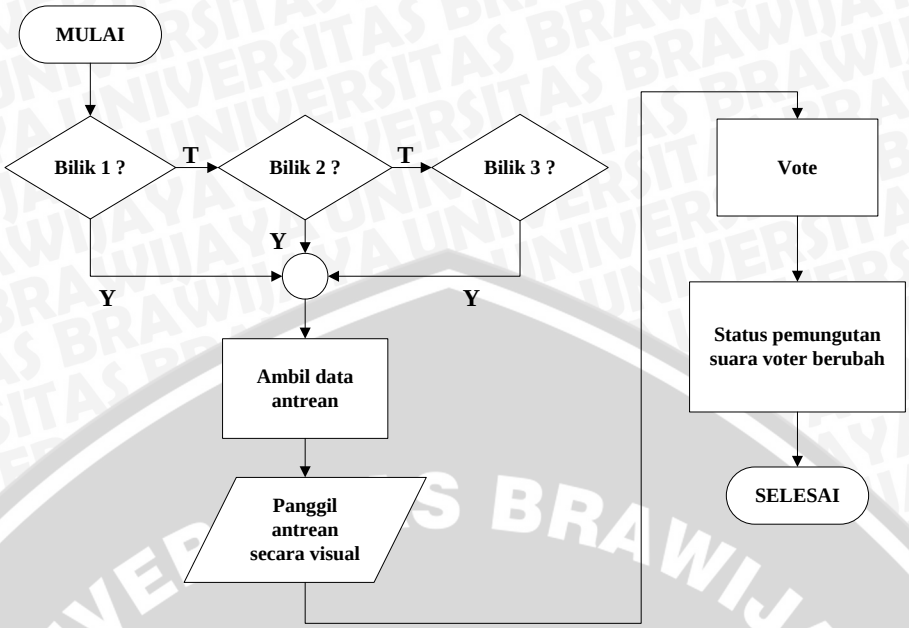
Panggilan ke	Pemanggilan visual		Pemanggilan audio
1	No. Antrean Nama Bilik	4 Adam Malik 2	*"Nomor empat Adam Malik menuju bilik 2"*
2	No. Antrean Nama Bilik	10 Soekarno 3	*"Nomor sepuluh Soekarno menuju bilik 3"*
3	No. Antrean	11	*"Nomor sebelas Nyi Ageng*

	Nama Bilik	Nyi Ageng Serang 2	*Serang menuju bilik 2"*
4	No. antrean Nama Bilik	70 Kartini 1	*"Nomor tujuh puluh Kartini menuju bilik 1"*
5	No. antrean Nama Bilik	85 Ahmad Yani 3	*"Nomor delapan puluh lima Ahmad Yani menuju bilik 3"*
6	No. antrean Nama Bilik	99 Dewi Sartika 2	*Nomor sembilan puluh sembilan Dewi Sartika menuju bilik 2"*
7	No. antrean Nama Bilik	100 Martha Christina 1	*"Nomor seratus Martha Christina menuju bilik 1"*
8	No. antrean Nama Bilik	191 Cut Nyak Meutia 2	*"Nomor seratus sembilan puluh satu Cut Nyak Meutia menuju bilik 2"*
9	No. antrean Nama Bilik	333 Mohammad Hatta 3	*"Nomor tiga ratus tiga puluh tiga Mohammad Hatta menuju bilik 3"*
10	No. antrean Nama Bilik	851 Soedirman 1	*"Nomor delapan ratus lima puluh satu menuju bilik 1"*

Dari keseluruhan hasil pengujian pemanggilan *voter* secara visual sama dengan hasil pengujian secara audio. Ini berarti bahwa terdapat kesesuaian data input di antara kedua sistem keluaran. Setelah dilakukan pengecekan dengan *database*, kedua jenis output telah sesuai dengan data yang ada pada tabel *database* antrean dan tabel *database voter*. Kesimpulan yang dapat diambil adalah bahwa program telah mampu memanggil *voter* sesuai dengan urutan antrean yang dibuat baik secara audio maupun visual dan mampu menampilkan data *voter* yang mendapat giliran antrean.

5.4 Pengujian Perubahan Status Pemungutan Suara *Voter*

Pengujian bertujuan untuk menganalisis kemampuan sistem dalam menyimpan data *voter* yang telah melakukan pemungutan suara. Pengujian dilakukan dengan memasukkan sepuluh data *voter* yang belum melakukan pemungutan suara dalam tabel *database* antrean. Kesepuluh data *voter* ini kemudian dipanggil secara visual menggunakan *software* yang dibuat melalui Delphi 7 dan disimulasikan untuk melakukan pemungutan suara secara sederhana. *Flowchart* pengujian ini ditunjukkan dalam Gambar 5.11.



Gambar 5.11 *Flowchart* pengujian perubahan status pemungutan suara *voter*

Untuk mengetahui perubahan status pemungutan suara selanjutnya dilakukan dengan memasukkan nomor ID dari *voter* yang diinginkan melalui program lanjutan setelah menekan tombol 'TES' pada dalam *software* yang dibuat. Selain itu, pengecekan status pemungutan suara *voter* juga dilakukan melalui tabel *database* antrean yang telah dibuat. Tabel *database* antrean yang digunakan untuk pengujian ini ditunjukkan dalam Gambar 5.12.

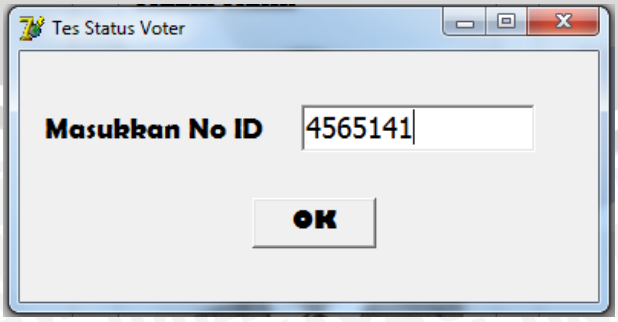
no	id	nama	bilik	status
4	4565141	Adam Malik	0	0
10	4565151	Soekarno	0	0
11	4565156	Nyi Ageng Serang	0	0
17	4374560	Cut Nyak Dhien	0	0
70	5931891	Kartini	0	0
85	4374555	Ahmad Yani	0	0
99	4565150	Dewi Sartika	0	0
100	5927687	Martha Christina	0	0
333	5931108	Mohammad Hatta	0	0
851	5931085	Soedirman	0	0

Gambar 5.12 Tabel *database* antrean yang digunakan sebagai acuan pengujian perubahan status pemungutan suara *voter*. Angka 0 pada kolom status berarti *voter* belum melakukan pemungutan suara.

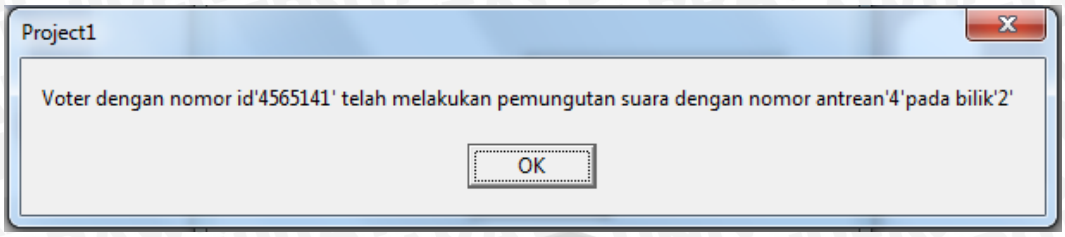
Pengujian yang pertama dilakukan dengan memanggil *voter* pada bilik 2. Setelah data *voter* ditampilkan secara visual seperti dalam Gambar 5.13,

disimulasikan *voter* melakukan pemungutan suara. Selanjutnya dilakukan pengecekan perubahan status *voter* melalui program lanjutan dengan melakukan pengisian nomor ID *voter* yang telah dipanggil sebelumnya, yakni 4565141 seperti ditunjukkan dalam Gambar 5.14. Setelah ditekan tombol 'OK' pada program tersebut, muncullah kotak dialog yang menyatakan bahwa *voter* telah melakukan pemungutan suara. Kotak dialog hasil pengecekan status *voter* tersebut ditunjukkan dalam Gambar 5.15. Dari pernyataan pada kotak dialog tersebut berarti status pemungutan suara *voter* pertama pada tabel *database* antrean yang sebelumnya masih belum melakukan pemungutan suara telah berhasil diubah. Hasil pengujian perubahan status pemungutan suara *voter* secara keseluruhan ditampilkan dalam Tabel 5.4, sedangkan tabel *database* antrean setelah dilakukan simulasi pemungutan suara ditunjukkan dalam Gambar 5.16.

Gambar 5.13 Tampilan pemanggilan *voter* yang pertama secara visual sebelum dilakukan simulasi pemungutan suara



Gambar 5.14 Tampilan program pengecekan status pemungutan suara *voter* yang pertama



Gambar 5.15 Tampilan kotak dialog hasil pengecekan perubahan status pemungutan suara *voter* yang pertama

Tabel 5.4 Hasil Pengujian Perubahan Status Pemungutan Suara *Voter*

***Voter* ke**	**Nomor ID**	**Nomor Urut Antrean**	**Status pemungutan suara**	
			Pada Program	**Pada *Database***
1	4565141	4	Telah melakukan pemungutan suara	1
2	4565151	10	Telah melakukan pemungutan suara	1
3	4565156	11	Telah melakukan pemungutan suara	1
4	4374560	17	Telah melakukan pemungutan suara	1
5	5931891	70	Telah melakukan pemungutan suara	1
6	4374555	85	Telah melakukan pemungutan suara	1
7	4565150	99	Telah melakukan pemungutan suara	1
8	5927687	100	Telah melakukan pemungutan suara	1
9	5931108	333	Telah melakukan pemungutan suara	1
10	5931085	851	Telah melakukan pemungutan suara	1

Catatan : Angka 1 untuk status pemungutan suara pada *database* berarti *voter* telah melakukan pemungutan suara

no	id	nama	bilik	status
4	4565141	Adam Malik	2	1
10	4565151	Soekarno	3	1
11	4565156	Nyi Ageng Serang	2	1
17	4374560	Cut Nyak Dhien	1	1
70	5931891	Kartini	3	1
85	4374555	Ahmad Yani	2	1
99	4565150	Dewi Sartika	1	1
100	5927687	Martha Christina	2	1
333	5931108	Mohammad Hatta	3	1
851	5931085	Soedirman	1	1

Gambar 5.16 Tabel *database* antrean setelah dilakukan simulasi pemungutan suara *voter*. Angka 1 pada kolom status berarti *voter* telah melakukan pemungutan suara

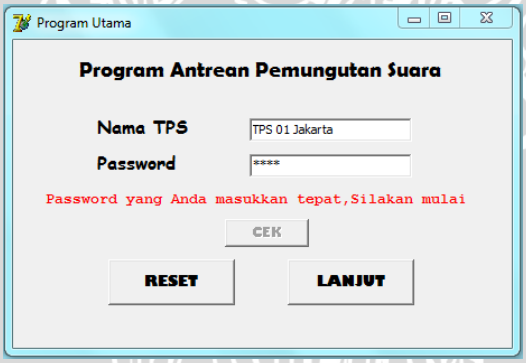
Dari hasil pengujian perubahan status *voter* secara keseluruhan terlihat bahwa seluruh status *voter* baik yang ditampilkan dalam program maupun yang disimpan di dalam *database* telah berhasil diubah. Hal ini berarti program telah

mampu untuk menyimpan hasil perubahan status *voter* yang telah melakukan pemungutan suara.

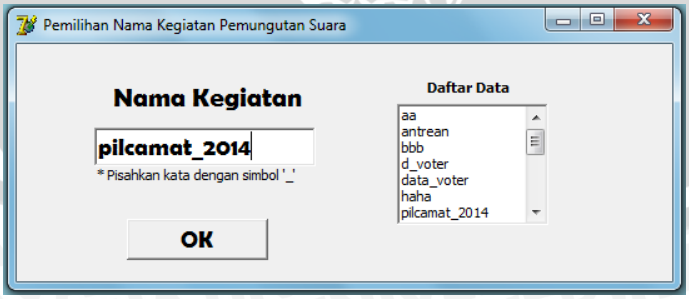
5.5 Pengujian Sistem Secara Keseluruhan

Pengujian sistem secara keseluruhan dimulai dari pengisian nama TPS dan *password* sistem lalu pengisian nama kegiatan pemungutan suara yang dilakukan dan selanjutnya adalah pengidentifikasian 5 buah tag RFID melalui RFID *Starter Kit* dan simulasi tag *voter* hilang. Hasil pembacaan dari RFID *Starter Kit* dan pengisian nama *voter* dengan tag hilang akan langsung memanggil data yang ada dalam *database* jika data ditemukan. Kemudian dilanjutkan dengan penyimpanan data dalam tabel antrean dan diakhiri dengan pemanggilan data melalui *speaker* dan monitor. Selanjutnya, selain disimulasikan *voter* melakukan pemungutan suara, dilakukan pula simulasi ketika *voter* yang dipanggil tidak kunjung hadir.

Urutan hasil pengujian sistem secara keseluruhan untuk pengisian nama TPS, *password* sistem dan pengisian nama kegiatan pemungutan suara ditunjukkan dalam Gambar 5.17(a) dan (b).



(a)

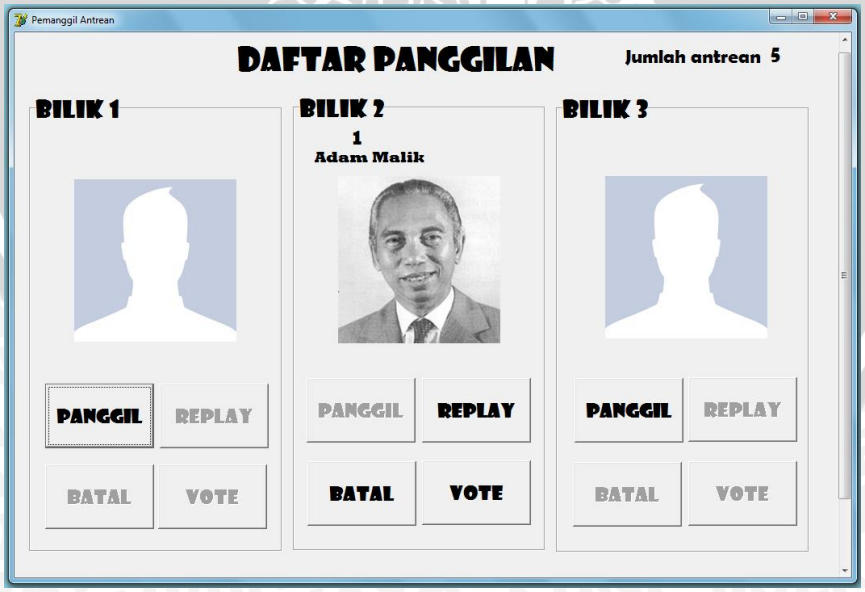


(b)

Gambar 5.17 (a) Tampilan pengujian pengisian nama TPS dan pengecekan *password* sistem
(b) Tampilan pengujian pengisian nama kegiatan pemungutan suara yang dilakukan

Dari hasil pengujian pengecekan *password* sistem menunjukkan bahwa sistem telah mampu menyeleksi benar tidaknya input *password* yang dimasukkan. Pengujian dilanjutkan dengan mengidentifikasikan 5 tag RFID *voter* yang berbeda dan sebuah nama lengkap *voter* dengan tag hilang. Selanjutnya dilakukan pemanggilan urutan antrean *voter* secara audio visual. Tampilan hasil pengujian pengidentifikasian data *voter* yang pertama setelah dimasukkan ke dalam antrean ditunjukkan dalam Gambar 5.18, sedangkan tampilan visual pemanggilan antrean *voter* yang pertama ditunjukkan dalam Gambar 5.19. Ouput audio dari pemanggilan *voter* yang pertama berbunyi "*Nomor satu Adam Malik menuju bilik 2*".

Gambar 5.18 Tampilan pengidentifikasian data *voter* yang pertama setelah dimasukkan ke dalam antrean

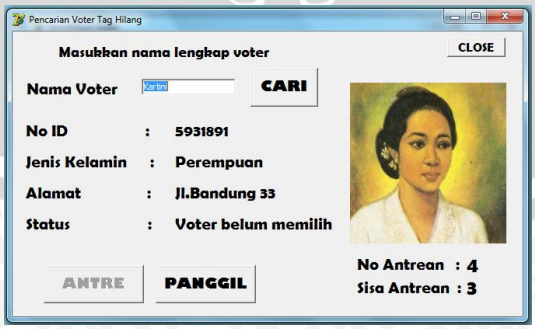


Gambar 5.19 Tampilan pemanggilan antrean *voter* secara visual yang pertama

Data yang ditampilkan pada pengidentifikasian data *voter* dan pemanggilan data *voter* secara audio dan visual yang pertama dicocokkan dengan tabel *database voter* yang ada dalam Gambar 5.5 sebelumnya. Hasilnya adalah keempat data tersebut ternyata telah sesuai. Tampilan pengidentifikasian data *voter* untuk *voter* kedua hingga keenam setelah dimasukkan ke dalam antrean ditunjukkan dalam Gambar 5.20 (a), (b), (c), (d), dan (e). Hasil pengujian keseluruhan pemanggilan *voter* secara audio visual beserta perlakuan yang diberikan ditampilkan dalam Tabel 5.5. Tabel *database* antrean *voter* setelah dilakukan pemanggilan secara keseluruhan ditunjukkan dalam Gambar 5.21.

(a)

(b)



(c)

(d)

(e)

Gambar 5.20 (a) Tampilan pengidentifikasian data *voter* kedua
(b)Tampilan pengidentifikasian data *voter* ketiga
(c) Tampilan pengidentifikasian data *voter* keempat dengan kartu tag hilang
(d) Tampilan pengidentifikasian data *voter* kelima
(e) Tampilan pengidentifikasian data *voter* keenam

Tabel 5.5 Hasil Pengujian Keseluruhan Pemanggilan Voter Secara Audio dan Visual

Panggilan ke	Pemanggilan visual		Pemanggilan audio	Perlakuan
1	No. antrean Nama Bilik	1 Adam Malik 2	*"Nomor satu Adam Malik menuju bilik 2"*	Vote
2	No. antrean Nama Bilik	2 Martha Christina 1	*"Nomor dua Martha Christina menuju bilik 1"*	Batal
3	No. antrean Nama Bilik	3 Pierre Tendean 3	*"Nomor tiga Pierre Tendean menuju bilik 3"*	Vote
4	No. antrean Nama Bilik	4 Kartini 2	*"Nomor empat Kartini menuju bilik 2"*	Vote
5	No. antrean Nama Bilik	5 Mohammad Hatta 1	*"Nomor lima Mohammad Hatta menuju bilik 1"*	Vote
6	No. antrean Nama	6 Dewi Sartika	*Nomor enam Dewi Sartika menuju*	Vote

	Bilik	2	*bilik 2”*	
7	No. antrean Nama Bilik	7 Martha Christina 3	*“Nomor tujuh* *Martha Christina* *menuju bilik 3”*	Vote

Keterangan : Vote : *voter* melakukan pemungutan suara ; Batal : *voter* belum memenuhi panggilan

no	id	nama	bilik	status
1	4565141	Adam Malik	2	1
3	4374549	Pierre Tendean	3	1
4	5931891	Kartini	2	1
5	5931108	Mohammad Hatta	1	1
6	4565150	Dewi Sartika	2	1
7	5927687	Martha Christina	3	1

Gambar 5.21 Tabel *database* antrean setelah dilakukan pemanggilan antrean secara keseluruhan

Dari keseluruhan pengujian pengidentifikasian *voter* dapat menampilkan data yang sesuai dengan tabel *database voter* yang ada. Untuk pengujian pemanggilan *voter*, pemangilan urutan *voter* telah sesuai dengan urutan penyimpanan antrean, di mana urutan tersebut sama dengan urutan pengidentifikasian *voter* melalui tag RFID. Ketika dilakukan simulasi ketidakhadiran *voter*, sistem telah mampu memindahkan nomor urutan *voter* pada urutan terbawah seperti yang terjadi pada *voter* Martha Christina yang awalnya memiliki nomor antrean 2 berubah menjadi nomor antrean 7. Dalam Gambar 5.21 diperlihatkan bahwa status pemungutan suara untuk semua *voter* telah berhasil diubah.

Dari keseluruhan hasil pengujian keseluruhan sistem ini menunjukkan bahwa sistem telah mampu mengidentifikasi *voter* melalui tag RFID, menampilkan data *voter* pada tabel *database* sesuai dengan nomor ID tag RFID, memanggil urutan antrean *voter* sesuai dengan data urutan antrean dan data *voter* yang tersimpan serta mampu mengubah status pemungutan suara *voter* setelah *voter* melakukan pemungutan suara.

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengujian tiap bagian sistem dan sistem secara keseluruhan, didapat kesimpulan sebagai berikut :

1) Perancangan dan pembuatan *software* antarmuka PC dengan RFID *Starter Kit* menggunakan komponen XcomDrv pada Delphi 7 telah berjalan dengan baik. PC mampu menampilkan data yang dibaca oleh RFID *Starter Kit* dengan tepat pada jarak baca maksimal 4,5 cm.
2) Perancangan dan pembuatan *software* PC penampil dan penyimpan data *voter* dalam antrean dilakukan dengan menghubungkan program tampilan yang dibuat melalui Delphi 7 dengan tabel *database* yang dibuat melalui MySQL. Perancangan dan pembuatan ini telah berjalan dengan baik, di mana data *voter* yang ditampilkan melalui program tampilan maupun yang disimpan di dalam tabel *database* antrean telah sesuai dengan data utama yang tersimpan di dalam tabel *database voter*.
3) Perancangan dan pembuatan *software* PC pemanggil antrean *voter* dilakukan dengan menghubungkan program tampilan (Delphi 7) dengan tabel *database* antrean *voter* (MySQL). Data dalam bentuk tulisan kemudian dikonversi menjadi data suara menggunakan fungsi *text to speech* dan *database* bahasa Inggris yang dibuat oleh Microsoft (Microsoft SAPI) pada Delphi 7 dan dikeluarkan melalui *speaker*. Perancangan dan pembuatan ini telah berjalan dengan baik, di mana terdapat kesesuaian urutan antrean dan data *voter* yang dipanggil (audio maupun visual) dengan data yang tersimpan di dalam *database*.
4) Pemberian tanda untuk *voter* yang telah melakukan pemungutan suara dilakukan dengan memberikan perubahan nilai status pemungutan suara *voter* pada tabel *database voter*. Nilai status itulah yang dijadikan sebagai kunci yang dapat menentukan dapat tidaknya *voter*

dalam melakukan pemungutan suara. Perancangan dan pembuatan ini telah berjalan dengan baik, di mana nilai status dari *voter* yang disimulasikan telah melakukan pemungutan suara dapat berubah dan disimpan pada *database*.

6.2 Saran

Saran – saran dalam pengimplementasian dan peningkatan kerja sistem ini dapat diuraikan sebagai berikut :

1) Sistem dapat dikembangkan dengan menambah jumlah kapasitas antrean lebih dari 999 nomor antrean.
2) Sistem dapat dikembangkan dengan menambahkan fasilitas pencetakan nomor antrean dan bukti telah melakukan pemungutan suara.
3) Sistem dapat dikembangkan dengan mengganti *text to speech* nama *voter* menggunakan ejaan Indonesia.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR PUSTAKA

Alkaf, Yasir. 2009. *Pemilu 2009 dalam Angka*. http://yasiralkaf.wordpress.com/2009/05/11/pemilu-2009-dalam-angka/. Diakses pada tanggal 14 Maret 2013 pukul 21 : 36.

Bolton, William. 2004. *Programmable Logic Controller (PLC) Sebuah Pengantar Edisi Ketiga*. Erlangga : Jakarta.

Budiharto, Widodo. 2008. *Panduan Praktikum Mikrokontroler AVR ATMega16, Panduan Utama untuk Pelajar dan Penghobi*. Elex Media Komputindo : Jakarta.

Huda, Miftakhul & Bunafit Komputer. 2010. *Membuat Aplikasi Database dengan Java, MySQL, dan NetBeans*. Elex Media Komputindo.

Hunt, V. Daniel, Albert Puglia dan Mike Puglia. 2007. *RFID : A Guide to Radio Frequency Identification*. John Wiley & sons : United States of America.

Ichwan, M. 2010. *Pemrograman Basis Data Delphi 7 dan MySQL Plus Penggunaan Paket Komponen Aplikasi Zeoslib, Scaladium dll*. Penerbit Informatika : Bandung.

Indoware. *125 kHz Low Frequency (LF) ISO Thin RFID Card TK4100 Passive.* http://indo-ware.com/produk-1976-125-khz-low-frequency-lf-iso-thin-rfid-card-tk4100-passive.html. Diakses pada tanggal 29 Juni 2013 pada pukul 10.03.

Innovations. 2005. *ID series Datasheet*. Mar 01,2005.

Innovative Electronics. 2007. Manual RFID Starter Kit.

J. Alam, M Agus. 2003. *Belajar Sendiri : Mengolah Database dengan Borland Delphi 7*. Elex Media Komputindo : Jakarta.

J. Alam, M. Agus. 2005. *Belajar Sndiri : Pemrograman Database Lokal dan Server Menggunakan Borland Delphi 2005*. Elex Media Komputindo : Jakarta.

Junerio, Adin. 2009. *Sistem Pemungutan Suara Menggunakan RFID dan Berbasis Komputer*. Skripsi. Malang : Program Sarjana Strata Satu Universitas Brawijaya.

Kadir, Abdul. 1999. *Konsep dan Tuntunan Praktis Basis Data*. Penerbit Andi : Yogyakarta.

Komisi Pemilihan Umum. 2009. *Buku Saku Pemilu 2009*. Jakarta Pusat.

Komputer, Wahana. 2010. *Membuat Aplikasi Client Server dengan Visual Basic 2008*. Penerbit Andi : Semarang.

Komputer, Wahana. 2010. *Panduan Belajar MySQL Database Server*. Media Kita : Jakarta.

Kusnassriyanto. 2011. *Belajar Pemrograman Delphi*. Penerbit Modula : Bandung.

McRoberts, Michael. 2010. *Beginning Arduino*. Technology in Action : United States of America.

Moussavi, Massoud. 2012. *Data Communication and Networking*. Delmar Cengage Learning : United States of America.

Mukhlasin, S. Kom, Hafid. 2008. *Cara Mudah Membuat Aplikasi Text To Speech Bahasa Inggris (TTS English) Menggunakan Delphi*. http://delphiscript.blogspot.com/2008/02/cara-mudah-membuat-aplikasi-text-to.html. Diakses pada tanggal 18 Maret 2013 pada pukul 07:30.

Musalini, Uus. 2004. *Membangun Aplikasi Super Cantik dan Full Animasi dengan Delphi*.. Elex Media Komputindo : Jakarta.

Nugroho, Bunafit. 2008. *Panduan Lengkap Menguasai Perintah SQL*. Media Kita : Jakarta.

Rozak, Mike. 1996. *Talk to Your Computer and Have It Answer Back with the Microsoft Speech API*. http://www.microsoft.com/msj/archive/S233.aspx. Diakses pada tanggal 5 Agustus 2013 pada pukul 11.03.

Sakur, Stendy B. 2005. *Aplikasi Web Database dengan Dreamweaver MX 2004*. Penerbit Andi : Yogyakarta.

Shelly, Cashman and Vermaat. 2007. *Discovering Computers : Menjelajah Dunia Komputer Fundamental, Edisi 3*. Penerbit Salemba Infotek.

Sunrom Technologies. 2012. *Datasheet USB to RS232 Serial UART Convertor – Plug*.

Supriyatno. 2010. *Pemrograman Database Menggunakan Java & MySQL Untuk Pemula*. Media Kita : Jakarta.

Syaukani, S.T., Muhammad. 2005. *Mengolah Data pada MySQL Server Menggunakan Visual FoxPro 8*. Elex Media Komputindo : Jakarta.

Undang – Undang Republik Indonesia Nomor 23 Tahun 2003 tentang Pemilihan Umum Presiden dan Wakil Presiden. 2003. Pustaka Widyatama : Yogyakarta.

W. Schuster, Edmund, Stuart J. Allen dan David L. Brock. 2007. *Global RFID : The Value of the EPCglobal Network for Supply Chain Management*. Springer : United States of America.

Want, Roy. 2006. *RFID Explained : A Primer on Radio Frequency Identification Technologies*. Morgan & Claypool Publisher.

Winda. 2009. *Pengenalan Radio Frequency Identification (RFID) dalam Kehidupan Sehari – Hari*. Skripsi. Jakarta : Program Sarjana Strata Satu Binus University.

Wulandari, Maulina. 2011. *Sistem Antrean pada Apotek Menggunakan Barcode dengan Output Audio – Visual*. Skripsi. Malang : Program Sarjana Strata Satu Universitas Brawijaya.

Yuhefizard, S. Kom. 2008. *Database Management Menggunakan Microsoft Access 2003*. Elex Media Komputindo : Jakarta.

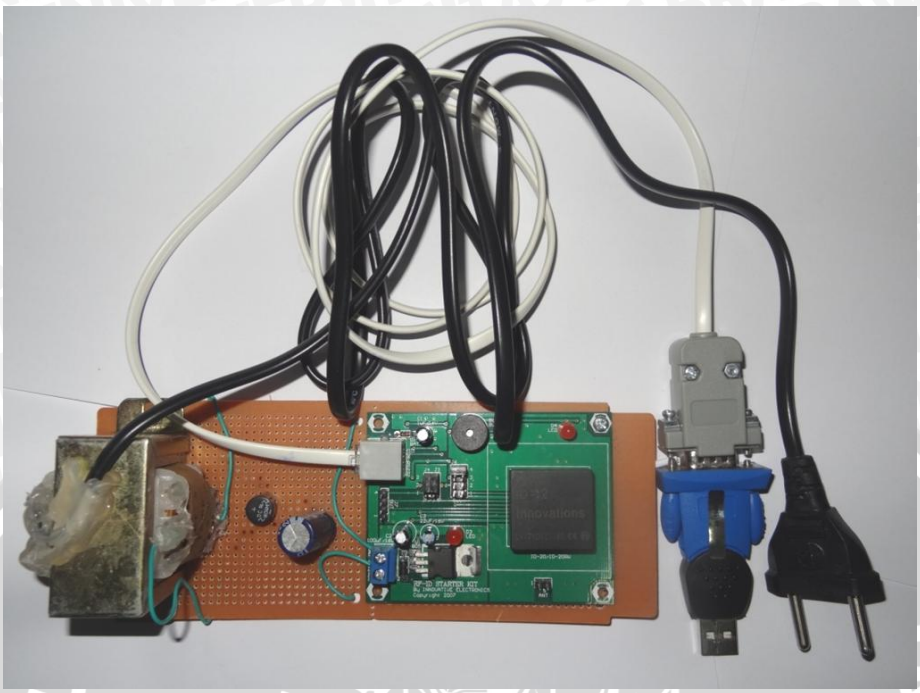
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

LAMPIRAN

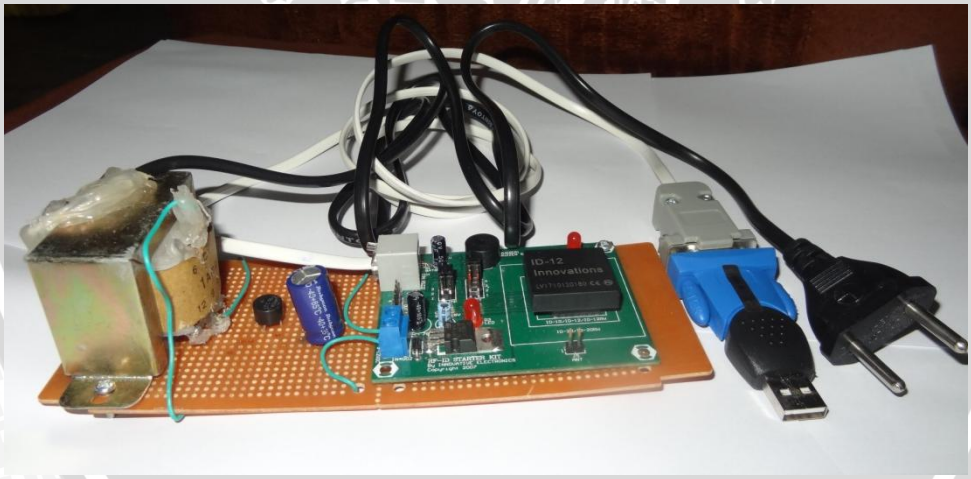
LAMPIRAN I

FOTO ALAT

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 1 Modul RFID *Starter Kit* dengan regulator catu daya dan serial to USB *converter* tampak atas



Gambar 2 Modul RFID *Starter Kit* dengan regulator catu daya dan serial to USB *converter* tampak samping

LAMPIRAN II

LISTING PROGRAM DELPHI 7

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
unit sampul;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TFutama = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    lbket: TLabel;
    edtps: TEdit;
    edpass: TEdit;
    cmdreset: TButton;
    cmdlanjut: TButton;
    cmdcek: TButton;
    cek: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure cmdcekClick(Sender: TObject);
    procedure cmdresetClick(Sender: TObject);
    procedure cmdlanjutClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Futama: TFutama;

implementation
 uses reset;
{$R *.dfm}

procedure TFutama.FormCreate(Sender: TObject);
begin
  lbket.Caption:='Masukkan nama TPS Anda dan password
sistem';
  cek.Caption:='';
  cmdcek.Enabled:=true;
  cmdreset.Enabled:=false;
  cmdlanjut.Enabled:=false;
end;

procedure TFutama.cmdcekClick(Sender: TObject);
var tps : string;
psw : integer;
begin
  tps := edtps.Text;
  psw:=StrToInt(edpass.Text);
  cmdcek.Enabled:=false;

  if psw = 1234 then
  begin
    cmdreset.Enabled:=true;
    cmdlanjut.Enabled:=true;
    lbket.Caption:='Password yang Anda masukkan
tepat,Silakan mulai';
  end
  else
  begin
    lbket.Caption:='Maaf Password Anda salah,sistem
tidak dapat dibuka';
    edpass.Text:='';
    cmdcek.Enabled:=true;
  end;
end;

procedure TFutama.cmdresetClick(Sender: TObject);
begin
cek.Caption:='0';
Freset.Show;
Futama.hide;
end;

procedure TFutama.cmdlanjutClick(Sender: TObject);
begin
```

```
cek.Caption:='1';
Freset.Show;
Futama.hide;
end;

end.
```

```
unit reset;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
  Dialogs, DB, ADODB, StdCtrls;

type
  TFreset = class(TForm)
    Label1: TLabel;
    edbaru: TEdit;
    cmdok: TButton;
    ADOConnection1: TADOConnection;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    ListBox1: TListBox;
    Label2: TLabel;
    Label3: TLabel;
    procedure cmdokClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Freset: TFreset;

implementation
  uses reader,sampul;
{$R *.dfm}

procedure TFreset.cmdokClick(Sender: TObject);
begin
  if edbaru.Text ='d_voter' then ShowMessage('Maaf,tabel
tersebut tidak bisa dibuka')
  else
  begin
    if Futama.cek.Caption='0' then
    begin
      ADOQuery1.Close;
      ADOQuery1.SQL.Clear;
      ADOQuery1.SQL.Add('CREATE TABLE '+edbaru.Text
      +'( no INT(2) PRIMARY KEY AUTO_INCREMENT, id
INT(15), nama VARCHAR(50), bilik INT(1),status
TINYINT(1))');
      ADOQuery1.ExecSQL;

      ADOQuery1.Close;
      ADOQuery1.SQL.Clear;
      ADOQuery1.SQL.Add('UPDATE d_voter SET
status=0,bilik=0,nomor_antrean=0,tps=NULL');
      ADOQuery1.ExecSQL;
    end;
    Frfid.Show;
    Freset.Close;
    end;
  end;

procedure TFreset.FormCreate(Sender: TObject);
begin
  ADOQuery1.Close;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.Add('SHOW TABLES');
  ADOQuery1.Open;
  ADOQuery1.First;

  ListBox1.Items.Clear;

  while not  ADOQuery1.eof do
    begin
      ListBox1.Items.add(ADOQuery1.fields[0].asstring);
      ADOQuery1.next;
    end;
end;

end.
```

```
unit reader;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, DB, ADODB, XComDrv;

type
  TFrfid = class(TForm)
    cmdpanggil: TButton;
    XComm1: TXComm;
    ADOConnection1: TADOConnection;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    lbket: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    lbno: TLabel;
    lbnama: TLabel;
    lbjk: TLabel;
    lbalamat: TLabel;
    lbstatus: TLabel;
    Image1: TImage;
    cmdantre: TButton;
    lbantre: TLabel;
    Label7: TLabel;
    cmdclose: TButton;
    Label1: TLabel;
    lbsisa: TLabel;
    cmdhilang: TButton;
    procedure cmdpanggilClick(Sender: TObject);
    procedure XComm1Data(Sender: TObject; const
Received: Cardinal);
    function hextoint(input : string):longint;
    procedure FormCreate(Sender: TObject);
    procedure ambil_data(nomor:string);
    procedure cmdantreClick(Sender: TObject);
    procedure cmdcloseClick(Sender: TObject);
    procedure cmdhilangClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Frfid: TFrfid;
   buffer : WideString;
   connect : boolean;
implementation
  uses panggil,reset,sampul,hilang;
{$R *.dfm}

function TFrfid.hextoint(input:string):longint;
var
    c,i : longint;
    input1 : string;
begin
    c:=0;
    input1:='';
    for i:=length(input) downto 1 do
input1:=input1+input[i]+'';
    input1:=uppercase(input1);
    for i:=1 to length(input1) do
    begin
     if (input1[i] in ['A'..'F'])and(i>1)then
        c:=((ord(input1[i])-ord('A')+10) shl (4*(i-
1)))or c
     else
     if (input1[i] in ['0'..'9'])and(i>1)then
        c:=((ord(input1[i])-ord('0')) shl (4*(i-1)))
or c
     else
     if (input1[i] in ['A'..'F'])and(i=1)then
        c:=(ord(input1[i])-ord('A')+10) or c
     else
     if (input1[i] in ['0'..'9'])and(i=1)then
```

```
          c:=(ord(input1[i])-ord('0')) or c;
    end;
    result:=c;
end;

procedure TFrfid.cmdpanggilClick(Sender: TObject);
begin
  Fpanggil.Show;
end;

procedure TFrfid.XComm1Data(Sender: TObject; const
Received: Cardinal);
var data,a : string;
begin
  XComm1.ReadString(data);
  buffer := buffer + data;
  if (data=#3) then
  begin
   a := copy(buffer,pos(buffer,#2)+4,8);
   lbno.Caption:=IntToStr(hextoint(a));

   ADOQuery1.Close;
   ADOQuery1.SQL.Clear;
   ADOQuery1.SQL.Add('SELECT * FROM d_voter WHERE id =
'+QuotedStr(lbno.Caption));
   ADOQuery1.Open;

   if ADOQuery1.IsEmpty then
   begin
    lbket.Caption:='Data voter tidak ditemukan';
    lbnama.Caption:='';
    lbjk.Caption:='';
    lbalamat.Caption:='';
    lbstatus.Caption:='';
    lbantre.Caption:='';
   end
   else ambil_data(lbno.Caption);
   buffer:='';
  end;
end;
```

```
procedure TFrfid.FormCreate(Sender: TObject);
begin
  lbket.Caption:= 'Dekatkan tag pada RFID Starter Kit';
  cmdantre.Enabled:=false;
  lbantre.Caption:='';
  lbsisa.Caption:='';
  image1.Picture.LoadFromFile('..\foto\default.bmp');

  if(connect = false) then
  begin
    XComm1.OpenDevice;
    connect:=true;
  end
  else
  begin
    XComm1.CloseDevice;
    connect := false;
  end;
end;

procedure TFrfid.ambil_data(nomor:string);
begin
    lbket.Caption:='Data ditemukan';
    lbnama.Caption :=ADOQuery1.FieldValues['nama'];
    if ADOQuery1.FieldValues['jenis_kelamin']=0 then
lbjk.Caption:='Perempuan'
    else lbjk.Caption:='Laki - Laki';
    lbalamat.Caption:=ADOQuery1.FieldValues['alamat'];

image1.Picture.LoadFromFile(ADOQuery1.FieldValues['foto'
]);
    if ADOQuery1.FieldValues['status']=0 then
    begin
      lbstatus.Caption:='Voter belum memilih';
      cmdantre.Enabled:=true;
    end
    else lbstatus.Caption:='Voter telah memilih';
end;

procedure TFrfid.cmdantreClick(Sender: TObject);
var antrean : string;
```

```
begin
  antrean := Freset.edbaru.Text;
  cmdantre.Enabled:=false;

  ADOQuery1.Close;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.add('SELECT * FROM '+antrean
  +' WHERE id ='+QuotedStr(lbno.Caption));
  ADOQuery1.Open;

  if ADOQuery1.IsEmpty then
  begin
     ADOQuery1.Close;
     ADOQuery1.SQL.Clear;
     ADOQuery1.SQL.Add('INSERT INTO '+antrean
     +' VALUES
(NULL,'+QuotedStr(lbno.Caption)+','+QuotedStr(lbnama.Cap
tion)+',0,0)');
     ADOQuery1.ExecSQL;

     ADOQuery1.Close;
     ADOQuery1.SQL.Clear;
     ADOQuery1.SQL.Add('SELECT no FROM '+antrean
     +' WHERE id ='+QuotedStr(lbno.Caption));
     ADOQuery1.Open;

     lbantre.Caption:= ADOQuery1.FieldValues['no'];

     ADOQuery1.Close;
     ADOQuery1.SQL.Clear;
     ADOQuery1.SQL.Add('SELECT COUNT(*) FROM '+antrean+
     ' GROUP BY status HAVING status = 0') ;
     ADOQuery1.open;

     lbsisa.Caption:=
(ADOQuery1.FieldValues['COUNT(*)'])-1;
  end
  else
  begin
     ShowMessage('Voter sudah masuk dalam antrean dengan
nomor '
     +QuotedStr(ADOQuery1.FieldValues['no']));
  end;
end;

procedure TFrfid.cmdcloseClick(Sender: TObject);
begin
  Futama.Close;
end;

procedure TFrfid.cmdhilangClick(Sender: TObject);
begin
  Fhilang.Show;
end;

end.
```

```
unit hilang;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
  Dialogs, DB, ADODB, StdCtrls, ExtCtrls;

type
  TFhilang = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    lbnama: TLabel;
    lbjk: TLabel;
    lbalamat: TLabel;
    lbstatus: TLabel;
    edid: TEdit;
    cmdcari: TButton;
    ADOConnection1: TADOConnection;
    DataSource1: TDataSource;
    ADOQuery1: TADOQuery;
    Image1: TImage;
    cmdantre: TButton;
    cmdpanggil: TButton;
    Label7: TLabel;
    lbantre: TLabel;
    cmdclose: TButton;
    lbket: TLabel;
    Label6: TLabel;
    lbsisa: TLabel;
    procedure cmdcloseClick(Sender: TObject);
    procedure cmdcariClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure cmdantreClick(Sender: TObject);
    procedure cmdpanggilClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Fhilang: TFhilang;

implementation
  uses panggil, reset, reader;
{$R *.dfm}

procedure TFhilang.cmdcloseClick(Sender: TObject);
begin
  Frfid.Show;
  Fhilang.Close;
end;

procedure TFhilang.cmdcariClick(Sender: TObject);
begin
  ADOQuery1.Close;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.Add('SELECT * FROM d_voter WHERE nama
='+ QuotedStr(edid.Text));
  ADOQuery1.Open;

  if ADOQuery1.IsEmpty then
  begin
    lbket.Caption:='Data voter tidak ditemukan';
    lbnama.Caption:='';
    lbjk.Caption:='';
    lbalamat.Caption:='';
    lbstatus.Caption:='';
    lbantre.Caption:='';
  end
  else
  begin
    lbnama.Caption := ADOQuery1.FieldValues['id'];
    if ADOQuery1.FieldValues['jenis_kelamin'] = 0 then
lbjk.Caption := 'Perempuan'
    else lbjk.Caption := 'Laki - Laki';
    lbalamat.Caption := ADOQuery1.FieldValues['alamat'];
```

```
    if ADOQuery1.FieldValues['status'] = 0 then
    begin
      lbstatus.Caption := 'Voter belum memilih';
      cmdantre.Enabled := true;
    end
    else lbstatus.Caption := 'Voter sudah memilih';

image1.Picture.LoadFromFile(ADOQuery1.FieldValues['foto'
]);
 end;
end;

procedure TFhilang.FormCreate(Sender: TObject);
begin
  lbket.Caption:= 'Masukkan nama lengkap voter';
  cmdantre.Enabled:=false;
  lbantre.Caption:='';
  lbsisa.Caption:='';
  image1.Picture.LoadFromFile('..\foto\default.bmp');
end;

procedure TFhilang.cmdantreClick(Sender: TObject);
var antrean : string;
begin
  antrean := Freset.edbaru.Text;
  cmdantre.Enabled:=false;

  ADOQuery1.Close;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.add('SELECT * FROM '+antrean
  +' WHERE id ='+QuotedStr(lbnama.Caption));
  ADOQuery1.Open;

  if ADOQuery1.IsEmpty then
  begin
     ADOQuery1.Close;
     ADOQuery1.SQL.Clear;
     ADOQuery1.SQL.Add('INSERT INTO '+antrean
     +' VALUES
(NULL,'+QuotedStr(lbnama.Caption)+','+QuotedStr(edid.Tex
t)+',0,0)');
     ADOQuery1.ExecSQL;

     ADOQuery1.Close;
     ADOQuery1.SQL.Clear;
     ADOQuery1.SQL.Add('SELECT no FROM '+antrean
     +' WHERE id ='+QuotedStr(lbnama.Caption));
     ADOQuery1.Open;

     lbantre.Caption:= ADOQuery1.FieldValues['no'];

     ADOQuery1.Close;
     ADOQuery1.SQL.Clear;
     ADOQuery1.SQL.Add('SELECT COUNT(*) FROM '+antrean+
     ' GROUP BY status HAVING status = 0') ;
     ADOQuery1.open;

     lbsisa.Caption:=
(ADOQuery1.FieldValues['COUNT(*)'])-1;
   end
   else
   begin
     ShowMessage('Voter sudah masuk dalam antrean dengan
nomor '
     +QuotedStr(ADOQuery1.FieldValues['no']));
   end;
end;

procedure TFhilang.cmdpanggilClick(Sender: TObject);
begin
  Fpanggil.Show;
  Fhilang.Close;
end;

end.
```

```
unit panggil;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
  Dialogs, DB, ADODB, MPlayer, StdCtrls, ExtCtrls;

type
  TFpanggil = class(TForm)
    Label1: TLabel;
    GroupBox1: TGroupBox;
    lbantre1: TLabel;
    lbnama1: TLabel;
    Image1: TImage;
    cmdpanggil1: TButton;
    cmdbatal1: TButton;
    cmdvote1: TButton;
    cmdulang1: TButton;
    GroupBox2: TGroupBox;
    lbantre2: TLabel;
    lbnama2: TLabel;
    Image2: TImage;
    cmdpanggil2: TButton;
    cmdbatal2: TButton;
    cmdvote2: TButton;
    GroupBox3: TGroupBox;
    lbantre3: TLabel;
    lbnama3: TLabel;
    Image3: TImage;
    cmdpanggil3: TButton;
    cmdbatal3: TButton;
    cmdvote3: TButton;
    cmdulang3: TButton;
    MediaPlayer1: TMediaPlayer;
    cmdulang2: TButton;
    ADOConnection1: TADOConnection;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    Label2: TLabel;
    lbjml: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure panggil(bil : integer);
    procedure batal (nomor :string);
    procedure konverter (input : string);
    procedure timer;
    procedure play_angka (angka : string);
    procedure replay (nomor,ident,bilik :string);
    procedure vote(name,antre,bil : string);
    procedure cmdpanggil1Click(Sender: TObject);
    procedure cmdulang1Click(Sender: TObject);
    procedure cmdbatal1Click(Sender: TObject);
    procedure cmdvote1Click(Sender: TObject);
    procedure cmdpanggil2Click(Sender: TObject);
    procedure cmdulang2Click(Sender: TObject);
    procedure cmdbatal2Click(Sender: TObject);
    procedure cmdvote2Click(Sender: TObject);
    procedure cmdpanggil3Click(Sender: TObject);
    procedure cmdulang3Click(Sender: TObject);
    procedure cmdbatal3Click(Sender: TObject);
    procedure cmdvote3Click(Sender: TObject);
    procedure jumlah;
    procedure FormShow(Sender: TObject);
    procedure puluhan (i,j,nomor : string);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Fpanggil: TFpanggil;
    SpVoice :Variant;
implementation
  uses comobj,sampul,reset;
{$R *.dfm}

procedure TFpanggil.jumlah;
var
noantre : string;
begin
```

```
  noantre:=Freset.edbaru.Text;
  ADOQuery1.Close;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.Add('SELECT COUNT(*) FROM '+noantre+
  ' GROUP BY status HAVING status = 0') ;
  ADOQuery1.open;

  if ADOQuery1.IsEmpty then
  else
  lbjml.Caption:= ADOQuery1.FieldValues['COUNT(*)'];
end;

procedure TFpanggil.FormCreate(Sender: TObject);
begin
  lbantre1.Caption := ' ';
  lbantre2.Caption := ' ';
  lbantre3.Caption := ' ';
  lbnama1.Caption := ' ';
  lbnama2.Caption := ' ';
  lbnama3.Caption := ' ';
  cmdbatal1.Enabled := false;
  cmdvote1.Enabled := false;
  cmdbatal2.Enabled := false;
  cmdvote2.Enabled := false;
  cmdbatal3.Enabled := false;
  cmdvote3.Enabled := false;
  cmdulang1.Enabled := false;
  cmdulang2.Enabled := false;
  cmdulang3.Enabled := false;
  image1.Picture.LoadFromFile('..\foto\default.bmp');
  image2.Picture.LoadFromFile('..\foto\default.bmp');
  image3.Picture.LoadFromFile('..\foto\default.bmp');
  SpVoice := CreateOleObject('SAPI.SpVoice');
end;

procedure TFpanggil.timer;
var i,j :integer;
begin
  for i:=0 to 40000 do
    for j:=0 to 40000 do;
end;
```

```
procedure TFpanggil.puluhan(i,j,nomor : string);
begin
  if i = '0' then
  begin
    if j <> '0' then
    begin
      timer;
      play_angka(j);
    end;
  end
  else if i='1' then
  begin
    if nomor = '10' then
    begin
      MediaPlayer1.FileName := '..\record\10.wav';
      MediaPlayer1.Open;
      MediaPlayer1.Play;
    end
    else if nomor= '11' then
    begin
      MediaPlayer1.FileName := '..\record\11.wav';
      MediaPlayer1.Open;
      MediaPlayer1.Play;
    end
    else
    begin
      play_angka(j);
      timer;

      MediaPlayer1.FileName := '..\record\belas.wav';
      MediaPlayer1.Open;
      MediaPlayer1.Play;
    end;
  end
  else
  begin
    play_angka(i);
    timer;

    MediaPlayer1.FileName := '..\record\puluh.wav';
```

```
    MediaPlayer1.Open;
    MediaPlayer1.Play;

    if j <> '0' then
    begin
      timer;
      play_angka(j);
    end;
  end;
end;

procedure TFpanggil.play_angka(angka:string);
var suara : variant;
begin
  if angka ='1' then suara := '..\record\1.wav'
  else if angka ='2' then suara := '..\record\2.wav'
  else if angka ='3' then suara := '..\record\3.wav'
  else if angka ='4' then suara := '..\record\4.wav'
  else if angka ='5' then suara := '..\record\5.wav'
  else if angka ='6' then suara := '..\record\6.wav'
  else if angka ='7' then suara := '..\record\7.wav'
  else if angka ='8' then suara := '..\record\8.wav'
  else if angka ='9' then suara := '..\record\9.wav'
  else suara := '..\record\kosong.mp3';

  MediaPlayer1.FileName := suara;
  MediaPlayer1.Open;
  MediaPlayer1.Play;
end;

procedure TFpanggil.konverter(input:string);
var angka : string;
i : integer;
begin
  angka :='';

  for i := 1 to length(input) do
    angka := angka + (input)[i]+'';

  if length(input)= 1 then play_angka(input)
  else if length (input)=2 then  puluhan
(angka[1],angka[2],input)
  else if length (input)=3 then
  begin
    if angka[1]='1' then
    begin
MediaPlayer1.FileName:='..\rekamaaaan\seratus.mp3';
      MediaPlayer1.Open;
      MediaPlayer1.Play;

      timer;
      puluhan (angka[2],angka[3],input);
    end
    else
    begin
      play_angka(angka[1]);
      timer;

      MediaPlayer1.FileName:='..\rekamaaaan\ratus.mp3';
      MediaPlayer1.Open;
      MediaPlayer1.Play;

      timer;
      puluhan(angka[2],angka[3],input);
    end;
  end;
end;

procedure TFpanggil.panggil(bil:integer);
var nomor,ident,ide,angka,gambar,noantre : string;
bilik : variant;
begin
  noantre:=Freset.edbaru.Text;
  ADOQuery1.Close;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.Add('SELECT
'+noantre+'.no,'+noantre+'.id,
d_voter.nama,d_voter.foto'
  +' FROM '+noantre+',d_voter WHERE '+noantre+'.id =
d_voter.id'
```

```
  +' GROUP BY '+noantre+'.status HAVING
'+noantre+'.status = 0');
  ADOQuery1.open;

  if ADOQuery1.FieldValues['no'] = NULL then
  begin
    ShowMessage('Maaf antrean masih kosong');
    if bil = 1 then
    begin
       cmdpanggil1.Enabled:=true;
       cmdbatal1.Enabled:= false;
       cmdulang1.Enabled:=false;
       cmdvote1.Enabled:=false;
    end
    else if bil = 2 then
    begin
       cmdpanggil2.Enabled:=true;
       cmdbatal2.Enabled:= false;
       cmdulang2.Enabled:=false;
       cmdvote2.Enabled:=false;
    end
    else
    begin
       cmdpanggil3.Enabled:=true;
       cmdbatal3.Enabled:= false;
       cmdulang3.Enabled:=false;
       cmdvote3.Enabled:=false;
    end
  end
  else
  begin
    mediaplayer1.FileName :='E:\record\nomor.wav';
    mediaplayer1.Open;
    mediaplayer1.Play;

    nomor:= IntToStr(ADOQuery1.FieldValues['no']);
    ident := ADOQuery1.FieldValues['nama'];
    gambar := ADOQuery1.FieldValues['foto'];

    if bil =1 then
    begin
      lbantre1.Caption := nomor;
      lbnama1.Caption := ident;
      Image1.Picture.LoadFromFile(gambar);
      bilik := '..\record\bilik 1.wav';
    end
    else if bil = 2 then
    begin
      lbantre2.Caption := nomor;
      lbnama2.Caption := ident;
      Image2.Picture.LoadFromFile(gambar);
      bilik := '..\record\bilik 2.wav';
    end
    else
    begin
      lbantre3.Caption := nomor;
      lbnama3.Caption := ident;
      Image3.Picture.LoadFromFile(gambar);
      bilik := '..\record\bilik 3.wav';
    end;

    timer;
    konverter (nomor);
    timer;
    SpVOice.speak (ident);

    mediaplayer1.FileName := bilik;
    mediaplayer1.Open;
    mediaplayer1.Play;

    angka := IntToStr(bil);
    ADOQuery1.Close;
    ADOQuery1.SQL.Clear;
    ADOQuery1.SQL.Add('UPDATE '+noantre+' SET status =
1,bilik = '+QuotedStr(angka)
    +'WHERE no ='+QuotedStr(nomor));
    ADOQuery1.ExecSQL;

    ADOQuery1.Close;
    ADOQuery1.SQL.Clear;
    ADOQuery1.SQL.Add('UPDATE d_voter SET status =
1,bilik = '+QuotedStr(angka)
```

```
    +'WHERE id ='+QuotedStr(ide));
    ADOQuery1.ExecSQL;
    jumlah;
  end;
end;

procedure TFpanggil.replay(nomor, ident, bilik :string);
begin
  mediaplayer1.FileName :='..\record\nomor.wav';
  mediaplayer1.Open;
  mediaplayer1.Play;
  timer;
  konverter (nomor);
  timer;
  SpVOice.speak (ident);

  mediaplayer1.FileName := bilik;
  mediaplayer1.Open;
  mediaplayer1.Play;
end;

procedure TFpanggil.batal(nomor:string);
var ide,noantre,name : string;
begin
  noantre:=Freset.edbaru.Text;
  ADOQuery1.Close;
  ADOQuery1.SQL.Clear;
  ADOQuery1.SQL.Add('SELECT id,no,nama FROM '+noantre+'
WHERE no = '+QuotedStr(nomor));
  ADOQuery1.open;

  if ADOQuery1.FieldValues['no'] <> NULL then
  begin
    ide := IntToStr(ADOQuery1.FieldValues['id']);
    name := ADOQuery1.FieldValues['nama'];

    ADOQuery1.Close;
    ADOQuery1.SQL.Clear;
    ADOQuery1.SQL.Add('DELETE FROM '+noantre+' WHERE id
='+QuotedStr(ide));
    ADOQuery1.ExecSQL;
    ADOQuery1.Close;
    ADOQuery1.SQL.Clear;
    ADOQuery1.SQL.Add('INSERT INTO '+noantre+' VALUES
(NULL,'+QuotedStr(ide)+','+QuotedStr(name)+',0,0)');
    ADOQuery1.ExecSQL;

    ADOQuery1.Close;
    ADOQuery1.SQL.Clear;
    ADOQuery1.SQL.Add('UPDATE d_voter SET status =
0,bilik = 0 WHERE id ='+QuotedStr(ide));
    ADOQuery1.ExecSQL;
  end;
  jumlah;
end;

procedure TFpanggil.vote(name,antre,bil:string);
begin
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('UPDATE d_voter SET tps =
'+QuotedStr(Futama.edtps.Text)
+', status = 1, bilik =' +QuotedStr(bil)+ ',
nomor_antrean ='
+QuotedStr(antre)+'WHERE nama ='+QuotedStr(name));
ADOQuery1.ExecSQL;
end;

procedure TFpanggil.cmdpanggil1Click(Sender: TObject);
var a : integer;
begin
  panggil(1);

  cmdpanggil1.Enabled:= false;
  cmdbatal1.Enabled := true;
  cmdvote1.Enabled := true;

  for a := 1 to 3 do timer;
  cmdulang1.Enabled := true;
end;
```

```
procedure TFpanggil.cmdulang1Click(Sender: TObject);
var nomor, ident, bilik : string;
b: integer;
begin
  cmdulang1.Enabled:= false;
  nomor := lbantre1.Caption;
  ident := lbnama1.Caption;
  bilik := '..\record\bilik 1.wav';

  replay(nomor,ident,bilik);

  for b := 1 to 2 do timer;
  cmdulang1.Enabled := true;
end;

procedure TFpanggil.cmdbatal1Click(Sender: TObject);
begin
  batal (lbantre1.Caption);
  lbantre1.Caption := ' ';
  lbnama1.Caption := ' ';
  image1.Picture.LoadFromFile('..\foto\default.bmp');

  cmdbatal1.Enabled := false;
  cmdvote1.Enabled := false;
  cmdpanggil1.Enabled := true;
  cmdulang1.Enabled := false;
end;

procedure TFpanggil.cmdvote1Click(Sender: TObject);
begin
  cmdvote1.Enabled := false;
  cmdpanggil1.Enabled := true;
  cmdulang1.Enabled := false;
  cmdbatal1.Enabled := false;
  vote(lbnama1.Caption,lbantre1.Caption,'1');
end;

procedure TFpanggil.cmdpanggil2Click(Sender: TObject);
var a:integer;
begin
  panggil(2);
  cmdpanggil2.Enabled:= false;
  cmdbatal2.Enabled := true;
  cmdvote2.Enabled := true;

  for a := 1 to 3 do timer;
  cmdulang2.Enabled := true;
end;

procedure TFpanggil.cmdulang2Click(Sender: TObject);
var nomor, ident, bilik : string;
b: integer;
begin
  cmdulang2.Enabled:= false;
  nomor := lbantre2.Caption;
  ident := lbnama2.Caption;
  bilik := '..\record\bilik 2.wav';

  replay(nomor,ident,bilik);

  for b := 1 to 2 do timer;
  cmdulang2.Enabled := true;
end;

procedure TFpanggil.cmdbatal2Click(Sender: TObject);
begin
  batal (lbantre2.Caption);
  lbantre2.Caption := ' ';
  lbnama2.Caption := ' ';
  image2.Picture.LoadFromFile('..\foto\default.bmp');

  cmdbatal2.Enabled := false;
  cmdvote2.Enabled := false;
  cmdpanggil2.Enabled := true;
  cmdulang2.Enabled := false;
end;

procedure TFpanggil.cmdvote2Click(Sender: TObject);
begin
  cmdvote2.Enabled := false;
  cmdpanggil2.Enabled := true;
  cmdulang2.Enabled := false;
```

```
  cmdbatal2.Enabled := false;
  vote(lbnama2.Caption,lbantre2.Caption,'2');
end;

procedure TFpanggil.cmdpanggil3Click(Sender: TObject);
var a : integer;
begin
  panggil(3);
  cmdpanggil3.Enabled:= false;
  cmdbatal3.Enabled := true;
  cmdvote3.Enabled := true;

  for a := 1 to 3 do timer;
  cmdulang3.Enabled := true;
end;

procedure TFpanggil.cmdulang3Click(Sender: TObject);
var nomor, ident, bilik : string;
b: integer;
begin
  cmdulang3.Enabled:= false;
  nomor := lbantre3.Caption;
  ident := lbnama3.Caption;
  bilik := '..\record\bilik 3.wav';

  replay(nomor,ident,bilik);

  for b := 1 to 2 do timer;
  cmdulang3.Enabled := true;
end;

procedure TFpanggil.cmdbatal3Click(Sender: TObject);
begin
  batal (lbantre3.Caption);
  lbantre3.Caption := ' ';
  lbnama3.Caption := ' ';
  image3.Picture.LoadFromFile('..\foto\default.bmp');

  cmdbatal3.Enabled := false;
  cmdvote3.Enabled := false;
  cmdpanggil3.Enabled := true;
  cmdulang3.Enabled := false;
end;

procedure TFpanggil.cmdvote3Click(Sender: TObject);
begin
  cmdvote3.Enabled := false;
  cmdpanggil3.Enabled := true;
  cmdulang3.Enabled := false;
  cmdbatal3.Enabled := false;
  vote(lbnama3.Caption,lbantre3.Caption,'3');
end;

procedure TFpanggil.FormShow(Sender: TObject);
begin
  jumlah;
end;

end.
```

LAMPIRAN III

MANUAL DAN *APPLICATION NOTE* RFID *STARTER KIT*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

LAMPIRAN IV

DATASHEET ID - 12

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA