IMPLEMENTASI PANEL PENGUNJUNG BERBASIS LAYAR SENTUH SEBAGAI SUB SISTEM KUNCI OTOMATIS PADA RUANG DOSEN UNIVERSITAS TELKOM

IMPLEMENTATION OF TOUCH SCREEN-BASED VISITOR PANEL AS A SUB SYSTEM OF AUTOMATIC LOCK DOOR AT TELKOM UNIVERSITY FACULTY ROOM

Muhammad Solihin[1], Agung Nugroho Jati[2], Umar Ali Ahmad[3]
[123]Prodi S1 Sistem Komputer, Universitas Telkom
[1]osoll73@gmail.com, [2]agungnj@telkomuniversity.ac.id, [3]uaa@ypt.or.id

Abstrak

Untuk kenyamanan dan kerahasiaan para dosen diperlukan sebuah sistem untuk membatasi akses masuk pada ruangan dosen. Sistem panel merupakan media bagi pengunjung untuk mengetahui keberadaan dosen di ruangan serta meminta kesediaan dosen agar bisa ditemui. Panel akan berupa monitor touch screen yang terhubung pada *raspberry pi* dan menampilkan aplikasi GUI. Sistem ini terhubung dengan sistem lainnya antara lain sistem mekanik kunci yang memberikan akses masuk kepada pengunjung dan dosen, dan sistem remote yang merupakan sarana bagi dosen untuk memberikan izin akses masuk kepada pengunjung. Sistem panel dan integrasi seluruh perangkat sudah berjalan dengan baik dan sesuai dengan yang diharapkan.

Kata Kunci : ***Visitor Panel, raspberry pi, Touch Panel***

Abstract

One of important aspect of working is comfortable place, no exception of the University lecturer. In addition of comfort, the other importance is secrecy. The two aspech can be preserved by either limiting people going to the room.Therefore, it is needed the system for that needing. This panel system is media for visitors to find out availability of recturer in the room and request the recturer to meet them. Panel is consist of touch screen monitor connected to raspberry pi and show a GUI application. This system is connected with other system including door mechanic system that provide access for entering the room, and remote system that is tool for the recturer to give access to the visitor. Panel system and the other integrated system have been succesfully running as expected.

Keywords: Visitor Panel, raspberry pi, Touch Panel

1. Pendahuluan

Salah satu aspek penting dalam bekerja adalah rasa nyaman, tidak terkecuali bagi para dosen Universitas. Selain kenyamanan, *privacy* juga merupakan aspek penting bagi mereka yang mengajar terutama kepada mahasiswa-mahasiswa yang mereka ajar. Pada hari-hari kerja, banyak orang-orang lalu lalang di ruangan yang dapat mengganggu kenyamanan serta *privacy* para dosen.

Oleh karena itu, diperlukan sistem yang dapat membatasi orang-orang selain dosen terutama mahasiswa memasuki ruang dosen. Sistem ini mengharuskan tamu atau pengunjung meminta izin untuk masuk atau menemui salah satu dosen melalu panel *monitor* yang disediakan di luar ruangan. Dosen akan menentukan akan mengizinkan atau tidak dengan menekan tombol di meja mereka. Jika dosen yang bersangkutan mengizinkan, maka pintu akan terbuka secara otomatis. Sistem panel ini juga akan memberitahukan ada atau tidaknya dosen yang bersangkutan di ruangan tersebut.

Dengan sistem ini, diharapkan kenyamanan serta privacy dosen dapat terjaga selama menjalankan tugasnya sehingga dapat tercipta proses ajar mengajar yang berkualitas. Sistem ini diharapkan dapat dikembangkan serta diimplementasikan pada ruang dosen yang bersangkutan dan ruang-ruang lainnya.

2. Dasar Teori

2.1. Raspberry Pi [4]

Raspberry Pi (atau Raspi) adalah sebuah komputer papan tunggal (SBC) berukuran kartu kredit yang dihubungkan ke TV (via HDMI) dan keyboard. Sebagai IoT (Internet of Things), seperti layaknya sebuah desktop. PC kecil ini mampu digunakan untuk menjalankan spreadsheet, pengolah kata dan permainan, terutama untuk memainkan video definisi tinggi.

Sistem operasi utama Raspberry Pi menggunakan Debian GNU/Linux, mengemas Iceweasel, kaligrafi Suite

dan bahasa pemrograman Python. Sejumlah distro lainnya, termasuk distro buatan Indonesia BlankOn Linux yang juga telah menyediakan versi arsitektur ARM dan bisa dijalankan di Raspi.

Hardware Raspberry Pi tidak memiliki real-time clock, sehingga OS harus memanfaatkan timer jaringan server sebagai pengganti. Namun komputer yang mudah dikembangkan ini dapat ditambahkan dengan fungsi real-time (seperti DS1307) dan banyak lainnya, melalui saluran GPIO (General-purpose input/output) via antarmuka I²C (Inter-Integrated Circuit).

2.2. Python [3]

Python merupakan Bahasa pemrograman yang freeware atau perangkat bebas dalam arti yang sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan source code-nya, debugger dan profiler, antarmuka yang terkandung didalamnya untuk pelayanan antarmuka, fungsi sistem, GUI (antarmuka penggunaan grafis), dan berbasis datanya. Python menjadi Bahasa resmi yang terintegrasi dalam Raspberry Pi. Kata "Pi" pada Raspberry Pi merupakan slang yang merujuk pada "Python". Oleh karenanya, tepat dikatakan bahwa Python adalah Bahasa natural Raspberry Pi.

Beberapa fitur yang dimiliki Python adalah:

1. Memiliki kepustakaan yang luas; dalam distribusi python telah disediakan modul-modul siap pakai untuk berbagai keperluan.
2. Memiliki tata Bahasa yang jernih dan mudah dipelajari.
3. Memiliki aturan layout kode sumber yang memudahkan pengecekan, pembacaan kembali, dan penulisan ulang kode sumber berorientasi objek.
4. Memiliki sistem pengelolaan memori otomatis (garbage collection seperti Java).
5. Modular, mudah dikembangkan dengan menciptakan modul-modul baru. Modul-modul tersebut dapat dibangun dengan Bahasa Python maupun C/C++.
6. Memiliki fasilitas pengumpulan sampah otomatis. Sperti halnya pada Bahasa pemrograman Java, Python memiliki fasilitas pengaturan penggunaan ingatan computer sehingga para pemrogram tidak perlu melakukan pengaturan ingatan computer secara langsung.

2.3. Flask [2]

Flask adalah *framework* aplikasi *web* mikro yang ditulis dalam bahasa *Python* dan berbasiskan *toolkit Wekzeug* dan *template engine Jinja2* dan berlisensi BSD. Pada tahun 2015, versi paling stabil *Flask* adalah versi 0.10.1. Contoh aplikasi yang menggunakan *framework Flask* adalah *Pinterest, LinkedIn,* dan tentu saja halaman *web Flask* itu sendiri.

Flask dikatakan framework mikro dikarenakan Flask tidak menganggap atau mengharuskan pengembang menggunakan alat atau pustaka tertentu. Flask tidak memiliki lapisan abstrak basis data, validasi form, dan komponen-komponen lainnya yang sudah dimiliki oleh pustaka-pustaka pihak ketiga sebelumnya. Walaupun begitu, Flask mendukung ekstensi yang dapat menambah fitur-fitur seperti layaknya mereka diimplementasikan di dalam Flask itu sendiri. Terdapat ekstensi untuk object-relational mappers, validasi form, upload handlint, dan berbagai teknologi otentikasi terbuka serta peralatan yang berhubungan dengan framework secara umum.

2.4. MariaDB [1]

MariaDB merupakan versi pengembangan terbuka dan mandiri dari *MySQL*. Sejak diakuisisinya *MySQL* oleh *Oracle* pada September 2010, Monty Program sebagai penulis awal kode sumber *MySQL* memisahkan diri dari pengembangan dan membuat versi yang lebih mandiri yakni *MariaDB*.

MariaDB adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MariaDB*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

2.5. SocketIO [5]

SocketIO merupakan *library javascript* untuk aplikasi *web realtime. SocketIO* memungkinkan komunikasi *bi-directional* yang *realtime* antara klien dan *server*. *SocketIO* memiliki dua buagian: *library* bagian klien dan *library* bagian server. Kedua komponen memiliki API yang indentik.

SocketIO menggunakan protokol WebSocket sebagai protokol utama dan polling sebagai opsi fallback, namun menggunakan antarmuka yang sama. Meskipun dapat digunakan sebagai wrapper untuk WebScoket, namun SocketIO menyediakan banyak fitur yang lain, antaranya broadcasting ke berbagai socket, menyimpan data yang berhubungan dengan setiap klien, dan asynchronus I/O.

3. Pembahasan

3.1. Gambaran Umum Sistem

Secara umum, sistem keseluruhan terdiri atas 3 sub sistem yaitu sistem mekanik pintu, sistem panel pengunjung, dan sistem remote. Sistem mekanik pintu akan berfokus pada sistem pintu yang akan otomatis terbuka dan terkunci pada saat ada *event* tertentu. Sistem panel akan memfasilitasi pengunjung yang datang untuk dapat meminta izin kepada dosen agar dapat memasuki ruangan. Sistem *remote* akan digunakan oleh dosen untuk memutuskan boleh atau tidaknya pengunjung untuk masuk ke dalam ruangan.

Pada penelitian ini, akan diimplementasikan sub sistem bagian Sistem Panel, dan *server* untuk komunikasi antar perangkat. Perangkat sistem panel terdiri dari komputer papan tunggal *Raspberry pi* yang berbagi dengan sistem pintu otomatis, dan sebuah LCD yang terpasang *touch screen* sebagai antarmuka pengunjung. *Server* dirancang untuk melayani ketiga jenis perangkat dalam berkomunikasi dan memberikan data yang dibutuhkan. Sistem panel, pintu otomatis, *remote*, dan *server* terhubung dalam satu jaringan dengan menggunakan *switch* dan *access*

3.2. Spesifikasi Perangkat Keras Panel

3.2.1. Raspberry Pi

Raspberry PI digunakan sebagai komputer yang memproses instruksi-instruksi sitem panel baik sistem jaringannya maupun sistem GUI-nya. Pada penelitian ini digunakan Raspberry Pi tipe B+ dengan spesifikasi sebagai berikut:

1. CPU 700 Mhz Low Power ARM1176JZFS
2. GPU Dual Core Video Core IV
3. Memori 512MB SDRAM
4. Video Output HDMI
5. USB 4x USB 2.0
6. Storage micro SD Card
7. Networking 10/100 BaseT Ethernet

3.2.2. LCD dan *Touch Screen*

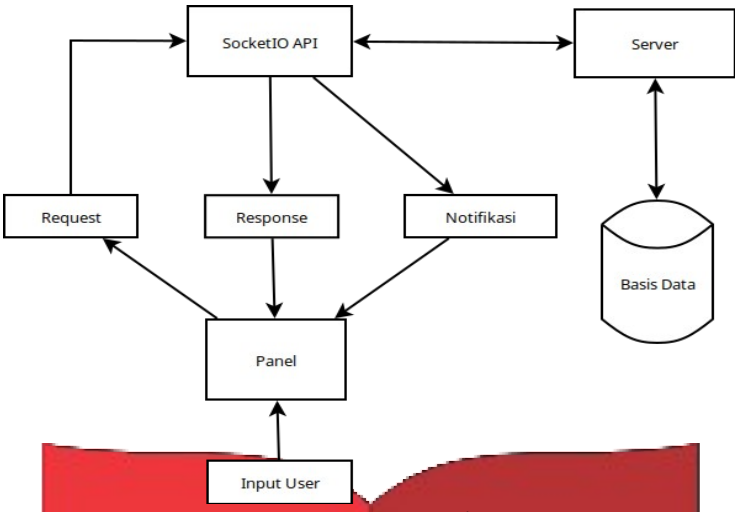
Untuk antarmuka dengan pengunjung dibutuhkan layar LCD dan *touch screen* sebagai alat inputnya. Pada penelitian ini, digunakan layar LCD sebesar 7 inchi dengan input HDMI dan telah terpasang *touch screen* dengan konektor USB. *Touch Screen* yang digunakan merupakan *touch screen* bertipe resistif, sehingga tidak seresponsif *touch screen* dengan tipe kapasitif.

3.3. Spesifikasi Perangkat Keras *Server*

Berikut perangkat keras *server* yang dibutuhkan

1. PC
   PC ini digunakan sebagai server yang menyediakan layanan API bagi perangkat-perangkat yang terhubung. PC yang digunakan merupakan komputer jinjing (notebook) dengan spesifikasi sebagai berikut:
   - Merk/Tipe Asus A43SA
   - Prosesor Intel® Core™ i5 2410M
   - Memori 4 GB
2. *Switch*
   Switch digunakan untuk perangkat komunikasi antara server dengan sistem panel.

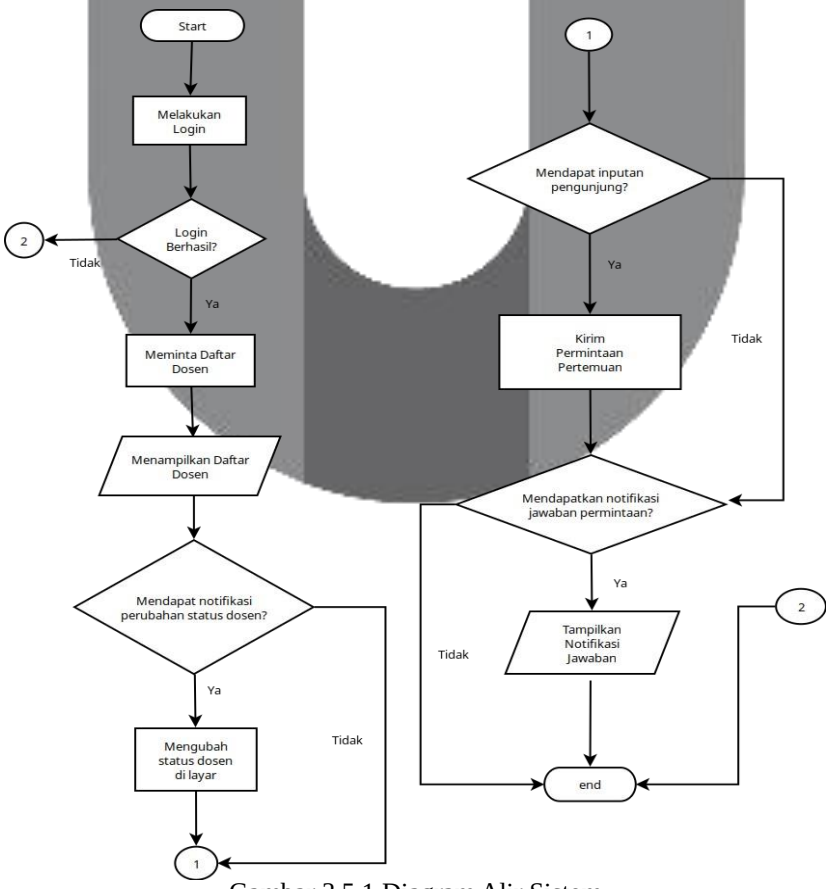
3.4. Skema Sistem



Adapun penjelasan d i t i l i i :

1. User me akukan inpu perm i an ma l l. antar muka pane l
2. Masukan yang yang diterima panel akan t skan ke API server melalui socketIO yang sudah terhubung sebelumnya.
3. Server akan mengolah permintaan dan mengolahnya
4. Server melakukan query ke basis data dan menyimpan permintaan
5. Server merespon berhasil atau tidaknya pembuatan permintaan

3.5. Diagram Alir Sistem



Gambar 3.5.1 Diagram Alir Sistem

3.6. Implementasi Panel

Panel digunakan sebagai perangkat penghubung antara pengunjung dan dosen. Pengunjung dapat melihat status ketersediaan dosen di dalam ruangan dan dapat melakukan permintaan pertemuan atau memasuki ruangan. Tampilan awal panel adalah daftar-daftar dosen beserta statusnya. Selanjutnya, pengunjung dapat memilih dosen yang ingin ditemui dengan menekan foto dosen tersebut.

Karena sistem mengenali pengunjung dari nomor serial rfid, maka sistem akan meminta pengunjung mendekatkan kartu yang memiliki *chip* rfid ke pembaca *rfid* yang berada di dekat panel. Pembaca *rfid* akan mengenali nomor serial lalu mengirimkannya ke *server*, lalu *server* mengirimkan kembali ke sistem panel. Panel akan melanjutkan ke *form* pengisian pengunjung, pengunjung diminta mengisinya lalu menekan tombol kirim.

3.7. Implementasi *Server*

Server menyediakan layanan berupa API yang digunakan perangkat-perangkat untuk mendapatkan data yang diinginkan. *Server* pula melakukan notifikasi otomatis kepada perangkat yang dianggap membutuhkan data yang telah dibuat atau diubah.

Perangkat dapat menggunakan layanan ini dengan cara login terlebih dahulu dengan id dan kata sandi yang tersimpan pada basis data. Perangkat dapat melakukan login dengan cara melakukan request POST ke server dengan protokol HTTP port 5000 dengan parameter id dan passwd. Setelah berhasil login, perangkat dapat melakukan koneksi ke SocketIO server port 5000 dengan namespace '/dev' dan dapat menggunakan API yang disediakan.

Server menyediakan layanan berupa API yang digunakan perangkat-perangkat untuk mendapatkan data yang diinginkan. *Server* pula melakukan notifikasi otomatis kepada perangkat yang dianggap membutuhkan data yang telah dibuat atau diubah.

3.8. Pengujian *Server*

Pengujian ini dilakukan untuk mengetahui kinerja server dalam menyediakan layanan API kepada perangkat-perangkat yang terhubung. Pengujian yang dilakukan adalah pengujian black box menggunakan beberapa skema pengujian yang sudah ditentukan, dan diujikan apakan keluaran yang dihasilkan sesuai dengan harapan atau tidak. Berikut hasil pengujian *server* dengan berbagai skenario yang diberikan:

Tabel 3.8.1 Hasil Pengujian *Server*

No.	Skenario pengujian	Hasil yang diharapkan	Kesimpulan
1	Perangkat melakukan login dengan *id* dan *password* yang benar	*Server* mendaftarkan *session* perangkat	Diterima
2	Perangkat melakukan login dengan *id* dan password yang salah	*Server* tidak mendaftarkan *session* perangkat, dan memutus hubungan dengan perangkat	Diterima
3	Panel terhubung	*Server* mendaftarkan perangkat ke daftar perangkat terhubung di memori, dan mengirimkan daftar dosen pada ruangan yang sama beserta statusnya	Diterima
4	Pintu terhubung	*Server* mendaftarkan perangkat ke daftar perangkat di memori	Diterima
5	*Remote t*erhubung	*Server* mendaftarkan perangkat ke daftar perangkat di memori, dan mengirimkan notifikasi data dosen ke panel untuk penggantian status dosen menjadi *online*	Diterima
6	Panel terputus	*Server* menghapus perangkat dari daftar perangkat di memori	Diterima
7	Pintu terputus	*Server* menghapus perangkat dari daftar perangkat di memori	Diterima
8	*Remote* terputus	*Server* menghapus perangkat dari daftar perangkat di memori, dan mengirimkan notifikasi data dosen ke panel untuk penggantian status dosen menjadi *offline*	Diterima
9	Pintu meminta akses masuk dengan nomor serial kartu rfid yang tidak/belum diizinkan memasuki ruangan	*Server* mengirimkan data *False* (tidak diizinkan)	Diterima
10	Pintu meminta akses masuk dengan nomor serial kartu rfid yang diizinkan memasuki ruangan	*Server* mengirimkan data *True* (diizinkan)	Diterima
11	Panel mengirimkan permintaan	*Server* menyimpan data permintaan ke basis data dan mengirimkan notifikasi ke *remote*	Diterima
12	*Remote* menjawab permintaan	*Server* mengubah data permintaan di basis data dan mengirimkan notifikasi ke panel	Diterima

Di sini juga dilakukan pengujian *load* server yaitu menguji seberapa banyak koneksi yang dapat *server* tangani dalam satu waktu. Pengujian ini dilakukan dengan mensimulasikan permintaan layanan *socketIO* untuk mendapatkan data dosen dengan menggunakan bahasa pemrograman *Python* dan teknik *multithreading* untuk mensimulasikan beberapa permintaan dalam satu waktu. Jumlah koneksi yang diuji adalah 50, 100, 150, 200, 250, dan 300 dengan masing-masing dilakukan 30 kali pengujian dan dihitung rata-ratanya. Berikut hasil pengujian *load server* yang dilakukan:

Tabel 3.8.2 Hasil Pengujian *Load Server*

Jumlah Koneksi	Jumlah Balasan	Total Waktu (detik)
50	50	0.4075
100	100	1.0187
150	150	1.6926
200	200	2.4539
250	250	2.268
300	300	2.5838

Pada pengujian jumlah koneksi 300, *server socketIO* mengalami masalah dan akhirnya *down.* Pengujian dengan jumlah koneksi 300 hanya sampai dengan 14 kali percobaan.

3.9. Pengujian Sistem Panel

Pengujian panel juga dilakukan dengan menggunakan pengujian *black box*, yaitu menguji beberapa fitur yang diberikan dengan parameter hasil yang diharapkan. Berikut hasil pengujian sistem panel dengan berbagai skenario yang diberikan:

Tabel 3.9.1 Hasil Pengujian Sistem Panel

No	Skenario pengujian	Hasil yang diharapkan	Kesimpulan
1	Panel login ke *server* dengan *id* dan *password* yang benar	Aplikasi berjalan dan menampilkan daftar dosen	Diterima
2	Panel login ke *server* dengan *id* dan *password* yang salah	Aplikasi tidak berjalan dan tidak menampilkan daftar dosen	Diterima
3	Panel terhubung sesudah beberapa *remote* terhubung	Panel menampilkan dosen dengan status beberapa dosen *online*	Diterima
4	*Remote* terhubung sesudah panel terhubung	Dosen berubah *online* pada tampilan dosen panel	Diterima
5	*Remote* terputus sesudah panel terhubung	Dosen berubah *offline* pada tampilan dosen panel	Diterima
6	Pengunjung memilih dosen dengan status *online*	Panel menampikan form perizinan akses masuk dengan *field* “kepada” berisikan nama dosen yang dipilih	Diterima
7	Pengunjung memilih dosen dengan status *offline*	Panel tidak berubah tampilan	Diterima

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dapat diambil kesimpulan bahwa implementasi panel berbasis *touch screen* sebagai penghubung pengunjung dengan para dosen pada suatu ruangan telah berhasil dibuat dengan perincian sebagai berikut:

1. Pengunjung dapat mengetahui ada atau tidaknya dosen dengan melihat daftar dosen yang tertera pada halaman utama panel
2. Pengunjung dapat meminta izin kepada dosen untuk memasuki ruangan dengan mengisikan *form* pada sistem panel dan sistem akan mengirimkannya ke *remote* dosen
3. Sistem panel dapat berkomunikasi dengan sub sistem lainnya melalui layanan *socketIO* API yang berada di *server*
4. *SocketIO* API mengalami *down* ketika menerima 350 koneksi secara bersamaan, yang dikarenakan aplikasi membuka terlalu banyak file *socket* di luar jumlah yang dibatasi sistem.

5. Waktu pemrosesan *request socketIO* rata-rata meningkat 0.498 detik per 50 koneksi

5. DAFTAR PUSTAKA

[1] Bartholomew, Daniel. 2013. *Getting Started with MariaDB. USA: Packtpub*
[2] Grinberg, Miguel. 2014. *Flask Web Development: Developing Web Applications with Python. USA: O'Reilly Media*
[3] *Lutz, Mark (2013). Learning Python, 5th Edition. USA: O'Reilly Media*
[4] Monk, Simon.(2014). *Raspberry Cook Book. USA: O'Reilly Media*
[5] Rohit, Rai (2013). *Socket.IO Real-time Web Application Development. USA: Packtpub*