

# SKRIPSI

## PEMANFAATAN QR CODE DALAM INPUT DATA ODOO, STUDI KASUS: SIMU



Dio Antares

NPM: 2017730003

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN

«tahun»



**UNDERGRADUATE THESIS**

**UTILIZATION OF QR CODES IN ODOO DATA INPUT SIMU  
CASE STUDY**



**Dio Antares**

**NPM: 2017730003**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY**

**«tahun»**



# LEMBAR PENGESAHAN

## PEMANFAATAN QR CODE DALAM INPUT DATA ODOO, STUDI KASUS: SIMU

Dio Antares

NPM: 2017730003

Bandung, «tanggal» «bulan» «tahun»

Menyetujui,

Pembimbing

Pascal Alfadian, Nugroho, M.Comp.

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng



## PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

### **PEMANFAATAN QR CODE DALAM INPUT DATA ODOO, STUDI KASUS: SIMU**

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,  
Tanggal «tanggal» «bulan» «tahun»



Dio Antares  
NPM: 2017730003





## **ABSTRAK**

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

**Kata-kata kunci:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»



## **ABSTRACT**

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

**Keywords:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»



*«kepada siapa anda mempersembahkan skripsi ini...?»*



## KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Bandung, «bulan» «tahun»

Penulis





# DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>xv</b>
<b>DAFTAR ISI</b>	<b>xvii</b>
<b>DAFTAR GAMBAR</b>	<b>xix</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	2
1.3 Tujuan . . . . .	2
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi . . . . .	2
1.6 Sistematika Pembahasan . . . . .	2
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 Odoo . . . . .	5
2.1.1 Struktur Direktori . . . . .	5
2.1.2 Instalasi . . . . .	6
2.2 Sistem Informasi Manajemen Umat (SIMU) . . . . .	9
2.2.1 Umat Baru . . . . .	10
2.2.2 Umat Pindah dari atau ke Paroki atau Lingkungan Lain . . . . .	10
2.2.3 Umat Masuk dari Keuskupan Lain . . . . .	11
2.2.4 Umat Keluar ke Keuskupan Lain . . . . .	11
2.2.5 Data Umat dan/atau Keluarga Berubah . . . . .	11
2.2.6 Umat Dibaptis . . . . .	12
2.3 Design untuk Aplikasi Mobile . . . . .	12
2.3.1 Pertimbangan Teknis . . . . .	12
2.3.2 User Interface Design . . . . .	13
2.3.3 Kesalahan Design Aplikasi Mobile . . . . .	13
2.4 QR Code . . . . .	13
2.4.1 Masalah pada Mempindai QR Code . . . . .	15
<b>DAFTAR REFERENSI</b>	<b>17</b>
<b>A KODE PROGRAM</b>	<b>19</b>
<b>B HASIL EKSPERIMEN</b>	<b>21</b>



## DAFTAR GAMBAR

2.1	Contoh Direktori Modul Odoo . . . . .	6
2.2	Contoh Intalasi Source menggunakan Git . . . . .	7
2.3	Contoh Melihat Versi Python dan Pip . . . . .	7
2.4	Contoh Membuat Database pada PostgreSQL . . . . .	8
2.5	Contoh Perintah untuk Melakukan Proses Instalasi Tambahan . . . . .	8
2.6	Contoh Perubahan Pengaturan pada PyCharm . . . . .	9
2.7	Contoh Halaman Odoo . . . . .	9
2.8	Contoh Formulir SIMU . . . . .	10
2.9	Contoh Versi QR Code . . . . .	14
2.10	Contoh Modul QR Code . . . . .	14
2.11	Contoh QR Code yang Terdistorsi . . . . .	15
2.12	Contoh QR Code yang DIkelilingi oleh Gambar atau Huruf . . . . .	15
2.13	Contoh QR Code yang Tumpang Tindih oleh Gambar atau Huruf . . . . .	16
B.1	Hasil 1 . . . . .	21
B.2	Hasil 2 . . . . .	21
B.3	Hasil 3 . . . . .	21
B.4	Hasil 4 . . . . .	21



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Pada saat ini kebutuhan manusia terhadap teknologi sangatlah tinggi, dapat dilihat dalam kehidupan sehari-hari manusia tidak terlepas dari penggunaan alat teknologi, karena dalam penggunaan teknologi dapat berfungsi sebagai alat untuk mempermudah melakukan sesuatu. Kemajuan teknologi yang kian pesat pada era modern ini membawa berbagai dampak pada banyak aspek kehidupan, termasuk dalam suatu organisasi. Sistem Informasi Manajemen Umat (SIMU) adalah aplikasi milik Keuskupan Bandung, aplikasi ini bertujuan untuk mencatat data umat dan dinamikanya (contohnya adalah sakramen). Keuskupan Bandung memiliki sekitar 108.000 umat, plus umat Sibolga.

Dengan banyaknya jumlah umat yang terdapat dalam sistem informasi dan tidak menutup kemungkinan akan terus bertambah, maka akan dibuat sebuah sistem dengan memanfaatkan perkembangan teknologi pada saat ini. Salah satunya adalah judul skripsi penulis pada saat ini yaitu Pemanfaatan QR Code dalam Input Data Odoo, Studi Kasus: SIMU. Pemanfaatan QR Code ini bertujuan untuk mempermudah, mempercepat proses input data dan mengurangi kesalahan penulisan dalam input data, karena data yang diinput sudah berdasarkan penulisan umat itu sendiri.

Sebelum sistem ini dibuat, maka jika perlu ada data umat yang dimasukkan ke sistem informasi manajemen umat (SIMU), prosedurnya adalah sebagai berikut:

1. Admin paroki memberikan blanko formulir data umat kepada umat.
2. Umat mengisi datanya ke dalam formulir tersebut secara tertulis.
3. Formulir dikembalikan kepada admin paroki.
4. Admin paroki mengetikkan data yang dituliskan di atas formulir.

Prosedur ini membutuhkan waktu yang lama dan kurang efisien, admin paroki memiliki kemungkinan untuk melakukan kesalahan dalam proses input data, karena admin paroki perlu untuk membaca ulang dan mengetikkan kembali data yang dituliskan di atas formulir kedalam sistem input data.

Pada skripsi ini yang berjudul Pemanfaatan QR Code dalam Input Data Odoo, Studi Kasus: SIMU, akan dibuat sebuah sistem yang dapat melakukan input data secara otomatis memanfaatkan QR Code pada input data Odoo, sehingga program yang dibuat ini akan menjadi dua aplikasi utama yaitu membuat halaman html sederhana (website) yang berisi form SIMU dan membuat sistem Odoo yang berisi data field yang menyerupai data umat SIMU dan sistem yang mampu memindai QR Code. Quick Response Code (QR Code) merupakan gambar dua dimensi yang memiliki kemampuan untuk menyimpan data. QR Code biasa digunakan untuk menyimpan data berupa teks, baik itu numerik, alfanumerik, maupun kode biner [1].

Program ini bertujuan agar umat dan admin paroki dapat lebih mudah dalam pengisian formulir, skripsi ini akan dirancang menggunakan Odoo. Odoo adalah aplikasi Enterprise Resource Planning open source berbasis Bahasa Python. Enterprise Resource Planning (ERP) adalah sebuah sistem informasi terintegrasi yang dapat mengakomodasi kebutuhan-kebutuhan informasi secara spesifik yang ada di perusahaan. Odoo adalah rangkaian aplikasi bisnis open source yang mencakup banyak kebutuhan, beberapa diantaranya adalah eCommerce, akuntansi, inventaris dan manajemen proyek.

## 1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas di skripsi ini adalah sebagai berikut :

1. Bagaimana membuat proses pencatatan data tidak perlu dilakukan secara manual?
2. Bagaimana agar data yang telah dituliskan oleh umat dapat dipindai oleh sistem SIMU?
3. Bagaimana agar data yang telah dituliskan oleh umat dapat diakses ke SIMU?

## 1.3 Tujuan

Tujuan yang ingin dicapai dari penulisan skripsi ini sebagai berikut :

1. Membangun halaman HTML yang responsif (terbaca mudah di ponsel) dan berisikan formulir sistem informasi manajemen umat (SIMU).
2. Membangkitkan kode QR berdasarkan data yang telah diisi untuk nantinya dibaca oleh Odoo.
3. Membangun sistem Odoo yang berisi data yang field-fieldnya menyerupai data umat dan sistem mampu memindai kode QR dari halaman formulir yang telah diisi oleh umat.

## 1.4 Batasan Masalah

Beberapa batasan yang dibuat terkait dengan pengerjaan skripsi ini adalah sebagai berikut :

1. Tidak mencoba Sistem Informasi Manajemen Umat (SIMU) yang asli, namun dilakukan percobaan melalui simulasi atau mockup sistem SIMU.

## 1.5 Metodologi

Metodologi yang dilakukan pada skripsi ini adalah sebagai berikut :

1. Melakukan studi literatur pembuatan modul Odoo.
2. Melakukan studi literatur User Experience yang baik di Mobile.
3. Menganalisis ukuran dan jumlah QR Code yang dibutuhkan.
4. Membangun sistem yang dapat dibuka di mobile dengan baik (responsive design), memunculkan keyboard yang tepat untuk input tertentu (contoh: nomor telepon menggunakan keypad), dan menyimpan data secara otomatis di penyimpanan lokal, sehingga saat dibuka kembali, umat dapat melanjutkan pengisian.
5. Melakukan pengujian dan eksperimen.
6. Menulis dokumen skripsi.

## 1.6 Sistematika Pembahasan

Sistematika penulisan setiap bab skripsi ini adalah sebagai berikut :

- Bab 1 Pendahuluan  
Membahas latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan.
- Bab 2 Landasan Teori  
Membahas teori-teori yang berhubungan dengan penelitian ini, yaitu Odoo, SIMU, dan QR Code.
- Bab 3 Analisis  
Membahas analisis terhadap sistem Odoo dan SIMU.
- Bab 4 Perancangan  
Membahas perancangan fitur yang akan diimplementasikan pada halaman website formulir dan SIMU.

- 
- Bab 5 Implementasi dan Pengujian  
Membahas implementasi fitur Odoo pada studi kasus SIMU dan pengujian yang dilakukan.
  - Bab 6 Kesimpulan dan Saran  
Membahas kesimpulan dari penelitian ini dan saran untuk penelitian berikutnya.





## BAB 2

### LANDASAN TEORI

#### 2.1 Odoo

Odoo adalah aplikasi *Enterprise Resource Planning* (ERP) open source adalah web aplikasi yang dibangun menggunakan bahasa pemrograman Python, XML, dan JavaScript dan menggunakan PostgreSQL sebagai database management sistemnya. Odoo merupakan sebuah sistem atau software manajemen open source, yang sangat mudah untuk digunakan. Bentuk dari sistem Odoo ini terdapat berbagai macam, diantaranya adalah berbasis web, desktop serta mobile. Selain itu, software ini memiliki banyak kelebihan seperti didukung oleh banyak komunitas, modul yang lengkap dan terintegrasi, pemasangan yang mudah, dan juga biaya yang terjangkau. Aplikasi bisnis yang terintegrasi dalam Odoo berbentuk modul-modul yang siap untuk diunduh dan digunakan dan sebagian besar bisa didapatkan secara gratis [2].

##### 2.1.1 Struktur Direktori

Pada bagian ini akan dibahas struktur direktori pada Odoo, salah satunya adalah Odoo Modul. Modul Odoo adalah perpaduan antara server dan client yang disatukan dalam satu modul yang dapat diakses atau dimuat melalui database. Modul Odoo adalah kumpulan fungsi dan data yang dapat melakukan berbagai hal dan tujuan. Segala sesuatu pada Odoo dimulai pada suatu modul. Penggunaan modul ini sendiri dapat dilakukan secara bebas sesuai dengan kebutuhan yang diperlukan oleh pengguna. Modul utama yang dapat dilihat oleh pengguna dapat berbentuk sebagai Aplikasi, namun sebagian besar modul bukan hanya Aplikasi. Modul juga dapat disebut sebagai addons dan direktori tempat server Odoo dapat menemukan addons yang dibuat oleh developer dapat dilihat pada folder addonspath <sup>1</sup>.

Komposisi pada modul dapat berisi sejumlah elemen, terdapat beberapa element yaitu:

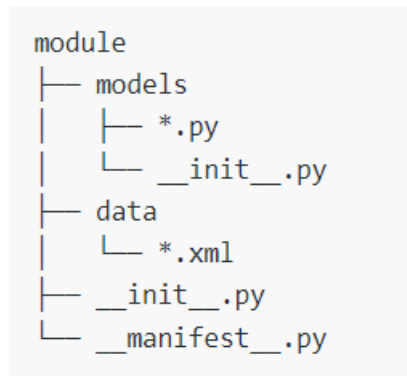
1. *Business Objects*  
Objek bisnis dideklarasikan sebagai kelas Python. Objek bisnis ini secara otomatis dipetakan ke kolom basis data.
2. *Object Views*  
Menampilkan tampilan *User Interface* (UI).
3. *Data Files*  
File XML atau CSV yang mendeklarasikan model data, beberapa contoh diantaranya adalah laporan, aturan keamanan, dan data demo.
4. *Web Controllers*  
Menanganin permintaan dari dari web browser.
5. *Static Web Data* File gambar, CSS, atau JavaScript yang digunakan oleh antarmuka website.

Struktur modul pada Odoo adalah direktori di dalam direktori modul. Direktori modul ditentukan dengan menggunakan opsi pada bagian folder `-addons-path`, dan modul Odoo dideklarasikan menggunakan file manifest. Ketika suatu modul akan dibuat maka modul tersebut diatur sebagai

---

<sup>1</sup>Modul Odoo [https://www.odoo.com/documentation/16.0/developer/tutorials/getting\\_started/01\\_architecture.html](https://www.odoo.com/documentation/16.0/developer/tutorials/getting_started/01_architecture.html)

sebuah file python dengan file `init.py`, file ini berisi instruksi impor untuk berbagai file python di dalam modul. Berikut adalah contoh direktori sebuah modul Odoo.



Gambar 2.1: Contoh Direktori Modul Odoo

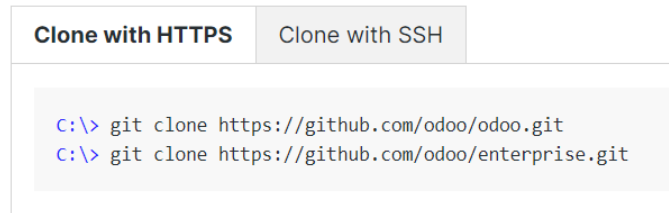
### 2.1.2 Instalasi

Pada skripsi ini Odoo akan diinstalasi menggunakan cara *Source Install*, proses instalasi ini bukan hanya sekedar install odoo dan menggunakannya langsung dari sumber website Odoo. Proses instalasi ini lebih nyaman digunakan oleh penulis karena untuk mengembangkan modul akan lebih mudah untuk diakses dibandingkan menggunakan instalasi yang sudah paket. Dalam proses penggunaan Odoo, akan lebih mudah untuk menjalankan dan menghentikan Odoo, sehingga terlihat lebih flexibel dibandingkan menggunakan instalasi yang sudah satu paket dan juga memungkinkan pengaturan menggunakan baris perintah, tanpa harus mengubah file konfigurasi pada Odoo. Secara tidak langsung, proses instalasi ini memberikan kontrol yang lebih besar atas pengaturan sistem dan memungkinkan untuk lebih mudah menyimpan dan menjalankan beberapa versi Odoo secara bersamaan.<sup>2</sup> Terdapat beberapa cara mengenai cara untuk melakukan instalasi Odoo 16, yaitu:

1. Online  
Instalasi secara online adalah cara termudah untuk menggunakan Odoo dalam membangun sistem produksi.
2. Package installer  
Instalasi secara *package installer* adalah cara yang sempurna untuk menguji Odoo, mengembangkan modul, dan dapat digunakan untuk penggunaan produksi jangka panjang dengan instalasi tambahan dan *maintenance* tambahan.
3. Install source  
Instalasi secara *install source* adalah cara install odoo dengan memberikan fleksibilitas yang lebih besar, contohnya adalah memungkinkan beberapa versi Odoo berjalan di sistem yang sama, baik untuk mengembangkan modul. Instalasi source ini adalah cara install Odoo yang akan digunakan pada skripsi ini.
4. Docker  
Instalasi Docker dapat digunakan untuk instalasi Odoo karena pengembangan aplikasi yang cepat, mudah, dan portabel.

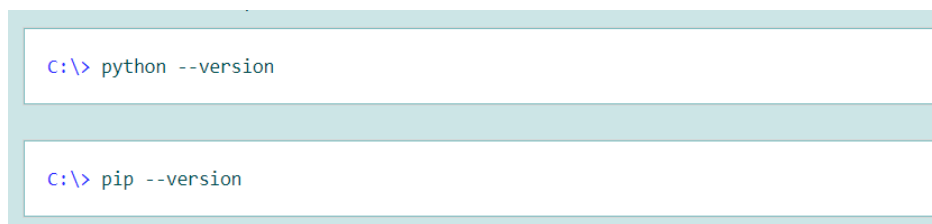
Pada proses instalasi secara source, terdapat dua cara untuk mengunduh kode Odoo, yaitu melalui arsip zip atau menggunakan git. Dalam penulisan skripsi ini akan dilakukan instalasi Odoo menggunakan git dalam mendapatkan kode Odoo. Tahapan pertama dalam instalasi ini adalah Git harus sudah terinstal di perangkat yang akan digunakan, dan developer harus memiliki pengetahuan dasar dalam proses penggunaan Git. Selanjutnya, untuk mengkloning repositori Git, developer harus memilih salah satu cara antara mengkloning dengan HTTPS atau SSH.

<sup>2</sup>Instalasi Odoo <https://www.odoo.com/documentation/16.0/administration/install/install.html>



Gambar 2.2: Contoh Intalasi Source menggunakan Git

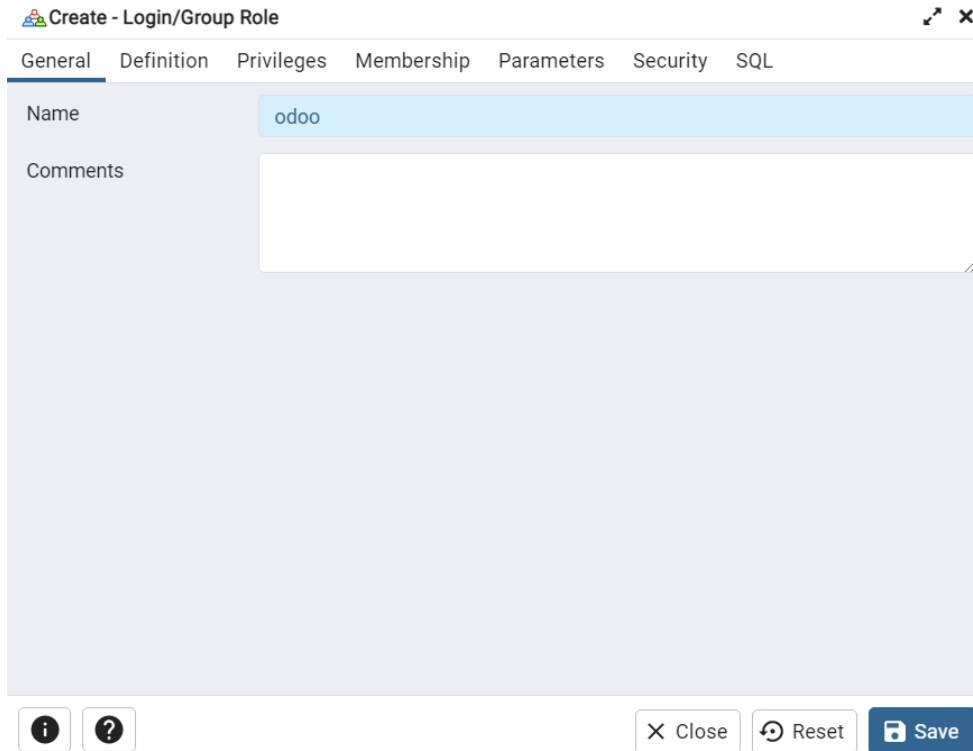
Tahap selanjutnya adalah mempersiapkan Python, sistem Odoo membutuhkan minimal versi 3.7 atau lebih, apabila Python sudah pernah diinstal, maka harus dilakukan pemeriksaan apakah Python sudah menggunakan versi 3.7 atau belum, karena versi dibawah 3.7 tidak cocok untuk instalasi Odoo. Cara yang dapat dilakukan untuk melihat versi Python dapat menggunakan cara sebagai berikut melalui *command prompt* (CMD).



Gambar 2.3: Contoh Melihat Versi Python dan Pip

Tahap selanjutnya adalah mempersiapkan PostgreSQL. Odoo menggunakan PostgreSQL sebagai sistem manajemen basis data. Pengguna dapat mengunduh dan instal PostgreSQL minimal versi 12.0 atau yang lebih terbaru. Pada proses instalasi PostgreSQL, pengaturan awal pengguna PostgreSQL adalah postgres, namun Odoo menyarankan untuk tidak menghubungkan database ke postgres, sehingga pengguna diharuskan untuk membuat user atau role baru di PostgreSQL. Berikut tahapan yang harus dilakukan ketika akan melakukan instalasi PostgreSQL:

1. Tambahkan direktori bin PostgreSQL (secara pengaturan awal tersimpan di C:-Program Files-PostgreSQL-<version>-bin) ke PATH perangkat yang digunakan. Pada penulisan skripsi ini, PostgreSQL yang digunakan adalah versi 15, sehingga penulisan pada path adalah (C:-Program Files-PostgreSQL-15-bin).
2. Buat baru nama pengguna postgres dengan kata sandi melalui pgAdmin GUI.
  - Buka program pgAdmin.
  - Klik dua kali pada bagian menu server untuk membuat koneksi.
  - Pilih bagian menu Objek lalu buat nama untuk login atau role.
  - Input nama di kolom nama (misalkan: odoo).
  - Pilih bagian *definition* lalu input password.
  - Pilih bagian *privileges* lalu pilih bagian dapat login dan buat database.



The screenshot shows a PostgreSQL 'Create - Login/Group Role' window. The 'General' tab is selected, displaying a form with a 'Name' field containing 'odoo' and a 'Comments' text area. The bottom of the window features three buttons: 'Close', 'Reset', and 'Save'.

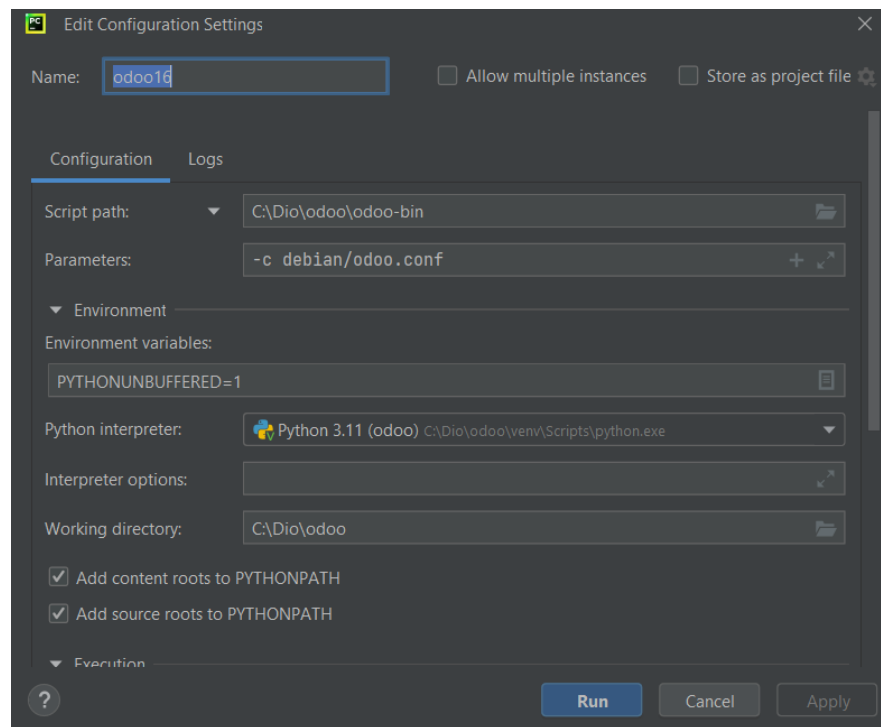
Gambar 2.4: Contoh Membuat Database pada PostgreSQL

Tahapan selanjutnya yang perlu dilakukan dalam instalasi Odoo adalah melakukan beberapa instalasi tambahan. Sebelum proses ini dilakukan, pengguna harus mengunduh dan menginstal *Build Tools for Visual Studio*, lalu pilih C++ build tools pada bagian tab Workloads dan lakukan proses instalasi. Setelah proses ini dilakukan, pengguna harus membuka *command prompt* (CMD) dan melakukan beberapa proses seperti pada gambar berikut:

```
C:\> pip install setuptools wheel
C:\> pip install -r requirements.txt
```

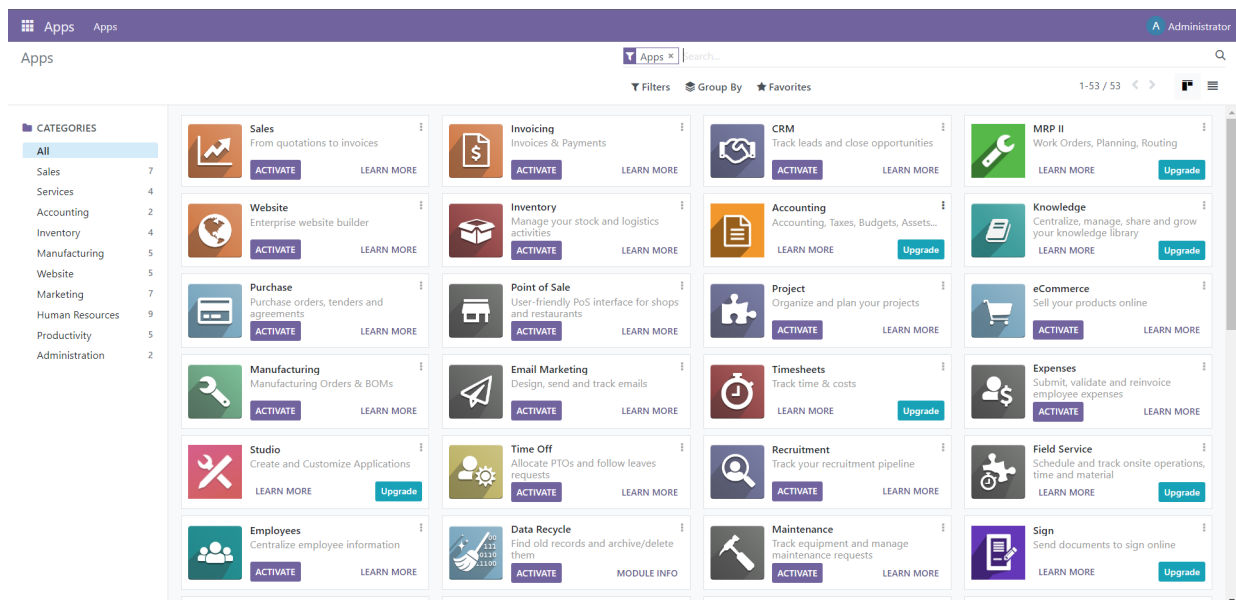
Gambar 2.5: Contoh Perintah untuk Melakukan Proses Instalasi Tambahan

Tahapan terakhir yaitu proses menjalankan Odoo, pada penulisan skripsi ini, penulis menggunakan aplikasi PyCharm, gunakan aplikasi ini untuk membuka folder yang sudah berhasil di clone lalu membukanya melalui PyCharm, setelah itu lakukan beberapa perubahan pada environment, sehingga server Odoo dapat dijalankan. Berikut contoh perubahan pada environment pada PyCharm:



Gambar 2.6: Contoh Perubahan Pengaturan pada PyCharm

Setelah server berhasil dijalankan (log INFO odoo.modules.loading: Modul sedang diproses), secara pengaturan awal, halaman untuk membuka website awal Odoo adalah <http://localhost:8069> yang dilakukan di browser web dan masuk dengan akun admin.



Gambar 2.7: Contoh Halaman Odoo

## 2.2 Sistem Informasi Manajemen Umat (SIMU)

Sistem Informasi Manajemen Umat (SIMU) adalah aplikasi milik Keuskupan Bandung, aplikasi ini bertujuan untuk mencatat data umat dan dinamikanya (contohnya adalah sakramen). Keuskupan Bandung memiliki sekitar 108.000 umat, plus umat Sibolga.

### 2.2.1 Umat Baru

Cara kerja sistem ini adalah apabila terdapat ada umat baru yang sebelumnya tidak tercatat di SIMU, maka admin akan memberikan print-out dari Formulir Data Umat kepada yang bersangkutan. Apabila keluarga belum tercatat di SIMU, maka admin akan memberikan print-out dari Formulir Keluarga Katolik atau Rumah Tangga Katolik untuk diisi. Formulir ini biasanya dimiliki oleh paroki masing-masing. Apabila tidak tersedia, maka umat dapat menghubungi admin keuskupan untuk mendapatkannya, dalam proses ini diharapkan umat dapat mengisi formulir dengan lengkap dan benar lalu dikembalikan ke sekretariat paroki.

Proses input data akan dilakukan oleh admin dengan cara admin memilih menu Umat, lalu admin akan klik tombol "Buat" di kiri atas, lalu data yang sudah ada akan diisikan ke dalam formulir, kemudian admin akan menyimpannya, dan untuk penulisan nama umat, umat diharapkan menuliskannya menggunakan huruf kapital secara keseluruhan. Apabila umat memiliki foto untuk dimasukan, umat dapat memasukan foto (opsional), semua hal tadi dapat diulangi oleh admin untuk seluruh umat baru yang akan dimasukkan datanya ke dalam sistem.

#### 1. Khusus Bayi

Apabila bayi yang baru lahir, umat diharapkan mengisikan "Belum Beragama" pada kolom agama, hal tersebut bertujuan supaya saat di masa depan akan menerima sakramen baptis, bayi tersebut muncul di daftar pilihan umat yang belum menjadi Katolik.

#### 2. Umat Ganda

Apabila sudah ada sistem deteksi umat ganda, admin diperlukan untuk memastikan bahwa umat belum pernah masuk sistem SIMU sebelumnya.

Gambar 2.8: Contoh Formulir SIMU

### 2.2.2 Umat Pindah dari atau ke Paroki atau Lingkungan Lain

#### 1. Seluruh Anggota Keluarga

Untuk memindahkan seluruh anggota keluarga ke paroki atau lingkungan baru, diperlukan prosedur sebagai berikut:

- Admin menari kepala keluarga dari keluarga katolik tersebut, kemudian klik “Pindah Paroki / Lingkungan”.
- Admin memastikan seluruh dokumen sudah diverifikasi (KTP, Surat Baptis, Surat Konfirmasi dari Ketua Lingkungan), lalu admin menekan klik seluruh checkbox yang disediakan, termasuk “Pindahkan seluruh anggota keluarga”. Klik “Simpan” untuk menyimpan.
- Admin mencari Keluarga Katolik dari umat tersebut. Setelah ditemukan, admin menekan klik Edit, dan sesuaikan kolom Paroki dan Lingkungan sesuai perubahan pada langkah

sebelumnya.

## 2. Salah Satu Anggota Keluarga

Untuk melakukan perpindahan umat sebagai salah satu anggota keluarga, diperlukan prosedur sebagai berikut:

- Admin mencari umat yang bersangkutan.
- Admin melakukan klik pada kolom “Paroki” dan atau “Lingkungan/KB”, dan mengisi nilainya dengan paroki tujuan.
- Jika umat tersebut berpindah karena menikah, maka umat tersebut harus dicabut dari keluarga yang lama dan dibuatkan atau dipindah ke keluarga baru, dengan cara admin melakukan edit, dan hapus umat tersebut dari keluarga tersebut melalui tab Anggota Keluarga, lalu admin membuat keluarga katolik baru melalui menu Umat dan mendaftarkan kedua anggota yang baru saja menikah (cukup satu keluarga per pasangan yang menikah).

### 2.2.3 Umat Masuk dari Keuskupan Lain

Dari Keuskupan yang Menggunakan BIDUK:

1. Umat melapor kepada admin SIMU paroki setempat
2. Admin Paroki SIMU berkoordinasi dengan admin SIMU Keuskupan melakukan permintaan atau request untuk menarik data umat yang bersangkutan dari BIDUK.
3. Admin SIMU Keuskupan masuk ke menu Catat Umat Masuk dan mengisi data umat baru tersebut.
4. Admin BIDUK menerima permohonan tarik data.
5. Admin BIDUK mengonfirmasi perpindahan keluar kepada admin paroki SIMU tentang keberadaan umat/keluarga yang dimaksud

### 2.2.4 Umat Keluar ke Keuskupan Lain

Menuju Keuskupan yang Menggunakan BIDUK, prasyarat dari proses ini adalah umat sudah berpindah secara tetap di paroki tujuan, langkah-langkahnya adalah sebagai berikut:

1. Umat melapor kepada admin BIDUK paroki setempat
2. Admin BIDUK melakukan permintaan atau request untuk menarik data umat yang bersangkutan dari SIMU
3. Admin SIMU menerima permohonan tarik data pada menu Mutasi Antar-Keuskupan Umat Keluar
4. Admin SIMU mengonfirmasi perpindahan keluar kepada admin paroki SIMU tentang keberadaan umat atau keluarga yang dimaksud.
5. Apabila perpindahan telah dikonfirmasi, admin SIMU menekan tombol “Setuju” pada permohonan mutasi tersebut.
6. SIMU akan otomatis mengirimkan data umat atau keluarga yang berpindah ke BIDUK, dan pada SIMU sendiri umat tersebut akan diset sebagai “non-aktif”.

Menuju Keuskupan Lain, pada prosedur ini tidak perlu menunggu umat yang bersangkutan untuk dikonfirmasi di keuskupan tujuan. langkah-langkahnya adalah sebagai berikut:

1. Umat melapor kepada admin paroki setempat
2. Admin Paroki berkoordinasi dengan admin Keuskupan melakukan permintaan atau request untuk menarik data umat yang bersangkutan dari BIDUK
3. Admin Keuskupan masuk ke Umat tersebut dan klik Pindah ke Keuskupan
4. Admin Keuskupan mengisi data yang diminta, dan menekan tombol “Simpan”.

### 2.2.5 Data Umat dan/atau Keluarga Berubah

Admin akan mencetak terlebih dahulu Formulir Data Umat dan Formulir Keluarga Katolik atau Rumah Tangga Katolik yang sudah terisi data SIMU, sehingga umat hanya perlu mengoreksi

informasi yang perlu diubah tanpa harus menuliskan ulang semuanya kembali. Setelah umat mengembalikan formulir yang sudah dikoreksi, admin akan melakukan perubahan data pada SIMU, dengan cara mencari kembali umat yang bersangkutan, dan memilih tombol Edit. Admin akan memperbaharui data-data yang berubah, kemudian admin akan menekan tombol “Simpan” untuk menyimpan perubahan.

### 2.2.6 Umat Dibaptis

Umat yang dibaptis harus sudah tercatat sebelumnya di SIMU. Apabila belum terdaftar maka:

1. Terdapat kemungkinan umat tersebut sudah didaftarkan di paroki lain. Dalam hal ini, umat harus berkoordinasi dengan paroki di mana umat tersebut berada.
2. Jika yakin bahwa umat tersebut belum terdaftar di SIMU, admin melakukan prosedur Umat Baru.

Persyaratan untuk melakukan baptis adalah umat yang akan dibaptis perlu melengkapi persyaratan seperti akte kelahiran, formulir calon baptis yang sudah diisi, dan sebagainya. Untuk setiap persyaratan yang telah dipenuhi, admin memberikan tanda centang pada tab “Persyaratan” (dengan sebelumnya membuka entri sakramen tersebut). Dalam proses baptis, calon baptis atau orang tua calon baptis juga perlu mendapatkan pendampingan. Jika pendampingan sudah selesai, umat akan melakukan “Edit” kembali entri yang bersangkutan, masuk ke tab “Pendampingan”, dan isikan tanggal kelulusan. Jika tanggal kelulusan sudah diisi, dan persyaratan lengkap, maka status akan bergerak maju menjadi “Persyaratan Terpenuhi”. Setelah proses materalisasi dilakukan, tahapan selanjutnya yang terakhir adalah surat-surat, setelah surat dicetak, belum tentu bisa langsung diambil oleh umat yang bersangkutan. Begitupun status di SIMU, di mana surat belum terambil.

## 2.3 Design untuk Aplikasi Mobile

Perangkat seluler (smartphone), tablet, game console telah menjadi hal yang umum di dunia komputasi. Desain seluler membuat tata letak estetika antarmuka pengguna. Desain seluler biasa dilakukan oleh software engineers, graphic designers, content developers, security specialists, dan semua orang yang tergabung dalam pembuatan model design. Desain sangatlah penting karena memungkinkan suatu model yang dibuat dapat meningkatkan nilai kualitasnya. Salah satu contohnya adalah website. Website adalah sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet. Untuk membangun sebuah halaman website dibutuhkan sebuah bahasa pemrograman yang lebih dikenal dengan sebutan web scripting. [?]

### 2.3.1 Pertimbangan Teknis

Pertimbangan teknis yang dilakukan untuk menurunkan biaya yang sangat rendah pada kapabilitas menambahkan web pada perangkat sehari-hari seperti ponsel, kamera, dan tv dapat mengubah cara orang mengakses informasi dan menggunakan layanan jaringan. Berikut merupakan beberapa pertimbangan teknis yang harus ditangani oleh aplikasi mobile:

1. Berbagai platform perangkat lunak dan keras  
Tidak bisa untuk produk yang berjalan diberbagai platform, karena terdapat perbedaan perangkat lunak dan keras sehingga banyak perbedaan diantara perangkat yang akan digunakan, dan akan membutuhkan waktu dan uang yang cukup mahal
2. Terlalu banyak frameworks dan bahasa pemrograman  
Banyaknya bahasa pemrograman dan frameworks yang digunakan membuat banyak perbedaan antara setiap perangkat mobile.
3. Terdapat banyak peraturan pada tempat publish aplikasi  
Setiap platform memiliki toko aplikasi dan standarnya sendiri untuk menerima aplikasi yang



dibuat. Sehingga setiap aplikasi mobile yang dibuat harus mengikuti setiap peraturan dan standar yang telah ada.

#### 4. Siklus Aplikasi Mobile

Pada siklus pengembangan aplikasi mobile, waktu yang dibutuhkan cukup lama dalam proses pembuatannya, namun pada akhirnya pasar persaingan aplikasi ini sangatlah cepat, sehingga apabila aplikasi tidak berkembang, maka aplikasi tersebut sudah dipastikan kalah oleh aplikasi lain yang terus bermunculan.

### 2.3.2 User Interface Design

Pengguna perangkat seluler berharap apabila mereka menggunakan aplikasi mobile, waktu belajar minimal yang diperlukan untuk mempelajari aplikasi tersebut diharapkan sangatlah cepat, oleh karena itu desainer aplikasi mobile harus bekerja keras dalam membuat suatu aplikasi. Berikut merupakan beberapa pertimbangan yang harus dilakukan dalam membuat *user interface design* pada aplikasi mobile:

1. Menentukan brand pokok dari produk tersebut, sehingga terdapat perbedaan dengan produk dari merk pesaing.
2. Fokus portofolio produk, menargetkan produk apakah untuk platform android atau ios, karena jumlah pengguna platform tersebut tidaklah sama.
3. Mengoptimalkan kecepatan dan kemampuan dari aplikasi yang dibuat, karena pengguna tidak mau banyak menunggu.
4. Tentukan ukuran dan scaling untuk produk yang akan dibuat, sehingga ketika menampilkan sesuatu tidaklah terlalu besar atau kecil.
5. Keahlian untuk melakukan design antarmuka harus sangat tinggi, karena untuk tata letak, animasi, grafik dibutuhkan keahlian khusus.

### 2.3.3 Kesalahan Design Aplikasi Mobile

Berikut merupakan beberapa kesalahan yang terdapat pada design aplikasi mobile:

1. Terlalu banyak fitur, hindari menambahkan banyak fitur yang kurang bermanfaat, karena hal tersebut akan mengurangi nilai keindahan, cukup sederhana namun bisa bersaing di pasaran.
2. Kurang konsisten, tentukan suatu standar pada produk aplikasi mobile yang akan dibuat, sehingga aplikasi yang akan dibuat memiliki patokan dan menghindari produk menjadi kurang konsisten.
3. Lag atau bisa dibilang kurang cepat dalam membuka atau melakukan sesuatu, hal seperti ini membuat pengguna menjadi banyak menunggu sehingga hanya membuang-buang waktu.
4. Design yang berlebihan, pemilihan warna, gambar, animasi, ataupun tema menjadi masalah yang penting, apabila produk tersebut memiliki design yang berlebihan, maka pengguna akan merasa tidak nyaman dengan tampilan yang ditampilkan.
5. Bertele-tele, aplikasi yang dibuat tidak sempat ditest, sehingga saat penggunaan aplikasi tersebut, banyak menu atau pilihan yang tidak berguna, sehingga tujuan yang akan dicapai oleh aplikasi tersebut menjadi hilang atau lama untuk tercapai.

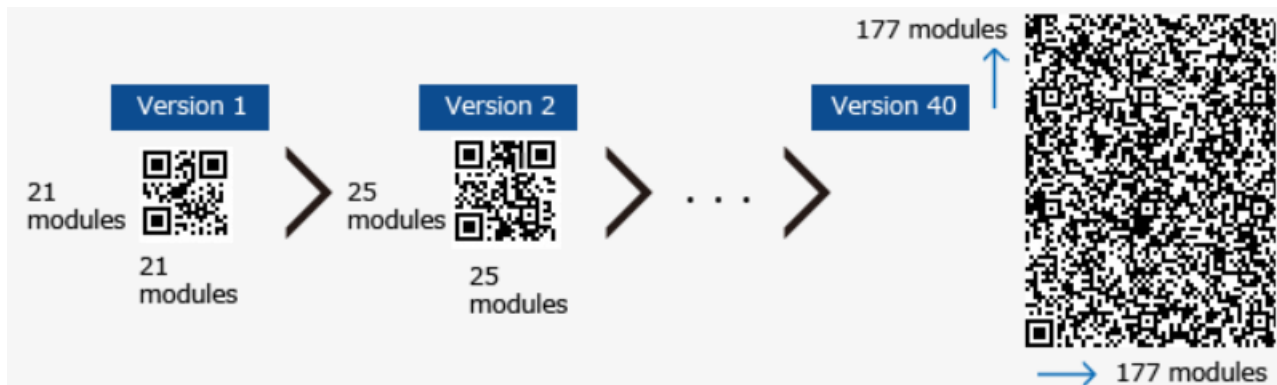
[?]

## 2.4 QR Code

QR Code, kependekan dari Quick Response Code, merupakan gambar dua dimensi yang memiliki kemampuan untuk menyimpan data. QR Code biasa digunakan untuk menyimpan data berupa teks, baik itu numerik, alfanumerik, maupun kode biner. QR Code banyak digunakan untuk keperluan komersil biasanya berisi link url ke alamat tertentu atau sekedar teks berisi iklan, promosi, dan lain-lain. QR Code adalah image dua dimensi yang merepresentasikan suatu data, terutama data

berbentuk teks. QR Code merupakan evolusi dari barcode yang awalnya satu dimensi menjadi dua dimensi. QR Code memiliki kemampuan menyimpan data yang lebih jauh besar daripada barcode.

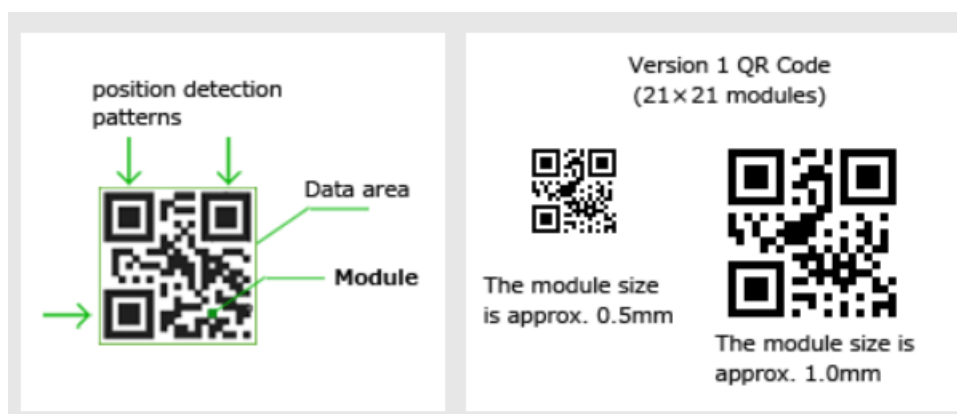
Versi simbol QR Code berkisar dari versi 1 sampai dengan Versi 40. Setiap versi memiliki konfigurasi modul atau jumlah modul yang berbeda, modul mengacu pada titik hitam dan putih yang membentuk QR Code. Konfigurasi modul ini dapat dilihat pada jumlah modul yang terdapat dalam simbol, biasa dimulai dengan versi 1 (modul  $21 \times 21$ ) hingga versi 40 (modul  $177 \times 177$ ). Setiap nomor versi yang lebih tinggi terdiri dari 4 modul tambahan per sisi.<sup>3</sup>



Gambar 2.9: Contoh Versi QR Code

Pada setiap versi simbol QR Code memiliki kapasitas data maksimum yang berbeda, tergantung dengan dengan jumlah data, jenis karakter, dan tingkat koreksi kesalahan. Oleh karena itu, seiring bertambahnya jumlah data, semakin banyak pula modul yang dibutuhkan untuk menyusun QR Code, sehingga menghasilkan simbol QR Code yang lebih besar.

Titik untuk mengukur ukuran sebenarnya dari simbol QR Code tergantung pada ukuran milimeter modul (satu area persegi yang terdiri dari QR Code) yang akan dicetak. Semakin besar modulnya, semakin stabil dan mudah dibaca dengan pemindai Kode QR Code. Namun dikarenakan ukuran simbol QR Code semakin besar, area pencetakan yang lebih besar akan diperlukan. Oleh karena itu, perlu untuk menentukan ukuran modul dari setiap aplikasi setelah mempertimbangkan semua faktor yang relevan. Namun tetap disarankan agar simbol QR Code dicetak sebesar mungkin dalam area pencetakan yang tersedia.



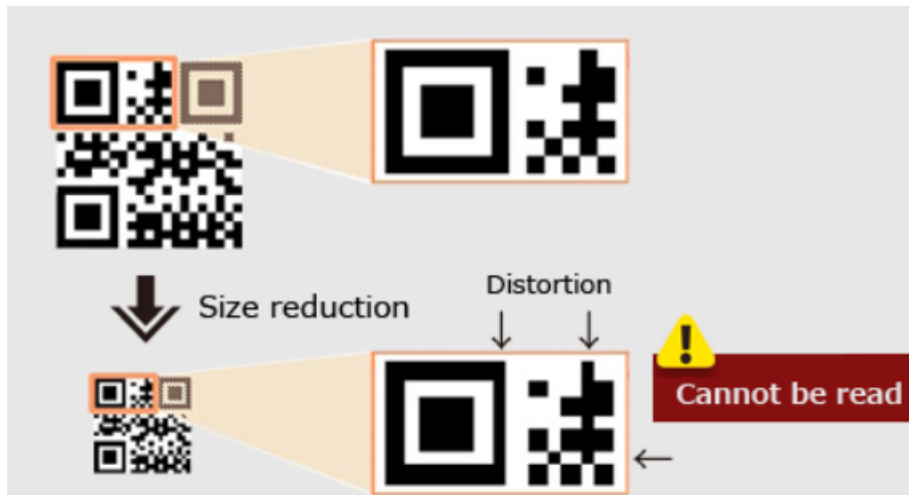
Gambar 2.10: Contoh Modul QR Code

<sup>3</sup>Versi Simbol QR Code <https://www.qrcode.com/en/about/version.html>

### 2.4.1 Masalah pada Mempindai QR Code

QR Code dapat dibaca dengan mudah apabila QR Code yang dibuat mengikuti standar QR Code dan harus dicetak dengan jelas. Oleh karena itu QR Code yang tidak mengikuti standar dan tidak jelas gambarnya, sudah dipastikan tidak dapat dibaca dengan jenis pemindai dan ponsel tertentu. Berikut contoh QR Code yang menyebabkan masalah dalam proses pemindaian QR Code:

1. QR Code yang modulnya terdistorsi, yang dimaksud disini adalah ketika QR Code diperbesar atau diperkecil menggunakan suatu aplikasi, maka QR Code tidak bisa dibaca atau dipindai.



Gambar 2.11: Contoh QR Code yang Terdistorsi

2. QR Code yang terdapat huruf atau gambar yang mengelilingi QR Code tersebut, hal ini menyebabkan camera atau alat scan menjadi sulit fokus terhadap modul QR Code.



Gambar 2.12: Contoh QR Code yang Dikelilingi oleh Gambar atau Huruf

3. QR Code yang tertimpa atau saling tumpang tindih oleh gambar atau huruf, akan menyebabkan kontras area antara warna gelap dan terang sulit untuk dibedakan.



Gambar 2.13: Contoh QR Code yang Tumpang Tindih oleh Gambar atau Huruf

## DAFTAR REFERENSI

- [1] Pasca, N. M. dan Rinaldi, M. (2011) Pengembangan aplikasi qr code generator dan qr code reader dari data berbentuk image. *Konferensi Nasional Informatika KNIF 2011*, **1**, 148–149.
- [2] Suminten (2019) Implementasi enterprise resource planning (erp) pada usaha pithik sambel ndesso berbasis odoo. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, **6**, 2597–9922.



# LAMPIRAN A

## KODE PROGRAM

Kode A.1: MyCode.c

```
1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf
```

Kode A.2: MyCode.java

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }
```





## LAMPIRAN B

### HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4