

BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Berikut merupakan kedudukan dan koordinasi penulis selama kerja magang di Sapphire Sky Hotel BSD.

3.1.1 Kedudukan

Pekerjaan magang di perusahaan ini, penulis bekerja di divisi *IT Information Technology*, saat ini memang perusahaan belum mendirikan departement *IT* sendiri sehingga divisi *IT* masih mengikuti *department Financial Controller* dalam kegiatan operasionalnya.

3.1.2 Koordinasi

Pada kesempatan kerja magang, banyak diskusi yang penulis lakukan dengan pembimbing yang memberikan tugas kepada penulis untuk bekerja dalam divisi *IT*. Diskusi tersebut dilakukan secara daring pada *text channel discord* untuk membahas pekerjaan yang telah diselesaikan penulis dan *review* yang diberikan pembimbing. Selain Diskusi, setiap bulan penulis mengadakan pertemuan *meeting* secara daring menggunakan *discord* untuk merepresentasikan hasil dan juga penulis diberikan solusi serta saran tentang masalah yang ingin dipecahkan.

Magang ini di supervisi dan diawasi secara langsung oleh bapak Adi Dharma, dimulai dari rancangan *UI* hingga alur dari program yang ingin dihasilkan, beberapa fitur yang ingin di fokus kan, serta memberikan pengarahan dalam membuat aplikasi agar dapat memenuhi ekspektasi.

3.2 Uraian Pelaksanaan Magang

Uraian kerja magang selama kerja magang penulis membuat sebuah aplikasi yang dapat memasukkan data pribadi *customer* yang melakukan *check-in* serta memasukkan data transaksi yang dilakukan pelanggan tersebut agar dapat mengetahui berapa *point* yang telah dikumpulkan oleh *customer* agar bisa mendapatkan *reward special* saat menginap. Selama seminggu penulis melakukan

coding selama lima hari dan biasanya ada pertemuan/meeting dengan *supervisor* untuk melakukan pengecekan terhadap aplikasi yang sedang dikerjakan.

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Jadwal Pekerjaan Magang Setiap Minggu

Minggu ke -	Pekerjaan yang dilakukan
1	<ul style="list-style-type: none"> • Perkenalan tentang perusahaan dan pekerjaan yang perlu dilakukan di perusahaan ini • Mencari bahasa pemrograman yang cocok dalam pembuatan aplikasi, berdiskusi tentang kekurangan serta kelebihan dari bahasa yang menjadi kandidat.
2	<ul style="list-style-type: none"> • Menentukan bahasa pemrograman yang digunakan dan sepakat untuk menggunakan bahasa pemrograman Python • Memahami penggunaan dasar <i>code editor</i> Pycharm dan penggunaan dari fitur, <i>packages</i>.
3	<ul style="list-style-type: none"> • Mempelajari tentang konsep dasar dari Python, pengenalan tentang beberapa variabel dan <i>library</i>. • Mencoba pembuatan aplikasi GUI <i>Graphical User Interface</i> dengan penggunaan <i>framework</i> Tkinter Python.
4	<ul style="list-style-type: none"> • Memikirkan rancangan dari <i>user interface</i> dari form tamu hotel dengan adanya menu simpel, menu <i>option</i> untuk menampilkan <i>window</i> yang baru. • Membuat menu simpel dibagian <i>form</i> dan menggunakan beberapa komponen <i>library framework</i> Tkinter <i>button</i> seperti <i>box checkbox</i> dan <i>message-box</i>, penggunaan label untuk menambahkan data tamu.

Tabel 3.1 Jadwal Pekerjaan Magang Setiap Minggu (lanjutan)

Minggu ke -	Pekerjaan yang dilakukan
5	<ul style="list-style-type: none"> • Menerapkan konsep CRUD (<i>Create, Read, Update, Delete</i>) ke dalam aplikasi GUI yang ingin dibuat . • Mempelajari lebih dalam mengenai <i>library</i> terbaru, mendeklarasikan variabel secara <i>global</i> dan <i>local</i>.
6	<ul style="list-style-type: none"> • Menggunakan database Mysql dan Sqlite3 untuk mengkoneksikan data tamu hotel yang <i>check-in</i> dengan konsep CRUD (<i>Create, Read, Update, Delete</i>).
7	<ul style="list-style-type: none"> • Memperbaiki model <i>database</i> yang masih <i>error</i> dan menambahkan fitur seperti <i>search</i>.
8	<ul style="list-style-type: none"> • Pembuatan <i>database profile</i> tamu dengan fitur <i>edit</i> spesifik terhadap <i>list</i> tamu yang ada sebagai antisipasi kesalahan dari <i>input</i>.
9	<ul style="list-style-type: none"> • Pembuatan <i>database</i> transaksi per tamu hotel yang menginap disertai dengan beberapa fitur seperti <i>CRUD record</i> dari masing masing tamu hotel.
10-12	<ul style="list-style-type: none"> • Membuat model <i>design</i> dari <i>database</i> transaksi riwayat menjadi <i>treeview</i>.
11-13	<ul style="list-style-type: none"> • Membuat aplikasi sistem <i>loyalt guest</i> untuk tamu yang menginap dengan adanya fitur <i>point</i> sebagai acuan dari <i>benefit</i> yang bisa didapatkan oleh tamu. • Mencari logika sistem <i>point loyalty guest</i> yang dapat di kalkulasi dari model kamar, jumlah kamar hingga lama tamu menginap. • Menambahkan fitur validasi <i>check-in</i> dan <i>check-out</i> untuk menentukan lama tamu menginap dan ter-hindar dari kesalahan <i>input</i>.

Tabel 3.1 Jadwal Pekerjaan Magang Setiap Minggu (lanjutan)

Minggu ke -	Pekerjaan yang dilakukan
14	<ul style="list-style-type: none"> • Penambahan data dengan sistem <i>child</i> dan <i>parent</i> dan iterasi selanjutnya dalam penggunaan <i>treeview</i>.
15-16	<ul style="list-style-type: none"> • Menyambungkan model <i>treeview</i> yang telah dibuat dengan <i>database</i> <i>sqlite3</i>. • Menggabungkan aplikasi secara keseluruhan dari <i>form database</i> tamu ke sistem <i>loyalty guest</i> menggunakan <i>database</i> detail transaksi tamu.
17-18	<ul style="list-style-type: none"> • Penambahan fungsionalitas seperti <i>button remove-all data</i>, <i>move up data</i>, <i>move down data</i>, dan <i>clear entry</i>. • Menambahkan fungsionalitas lain untuk sistem <i>point button</i> untuk mendapatkan total hari tamu menginap yang didapatkan dari <i>pick calender check-in</i> dan <i>check-out</i>.
19-21	<ul style="list-style-type: none"> • Melakukan perbaikan <i>bug</i> yang ditemukan di aplikasi <i>loyalty app</i>.

Tabel yang telah dibuat dalam bentuk *longtable* di atas menggambarkan secara singkat pekerjaan yang penulis lakukan selama melakukan magang di perusahaan Sapphire Sky Hotel BSD, di tabel tersebut pekerjaan penulis menjabarkan *point* penting pekerjaan yang dilakukan setiap minggu nya hingga akhirnya dapat selesai mengerjakan magang.

Dimulai dengan minggu pertama, penulis bersama teman kantor berkenalan dengan pembimbing atau *supervisor* untuk membahas permasalahan yang ada sehingga penulis ditugaskan untuk membuat program yang dapat menampung data tamu hotel yang menginap yang nantinya dapat di temukan secara mudah jika diperlukan. Penulis dan teman yang telah sepakat untuk mengerjakan tugas yang telah diberikan langsung mencari bahasa pemograman yang dapat menjadi dasar dalam pembuatan aplikasi dan penulis melakukan *brainstorming* untuk menemukan yang tidak hanya cocok namun mudah dipahami dan pernah penulis pakai sebelumnya di

project mata kuliah Informatika.

Pada minggu kedua, penulis sepakat dengan teman kantor untuk menentukan bahasa pemrograman bernama Python, penulis memilih Python karena mudah dipelajari, memiliki banyak *library* sehingga program yang dihasilkan dapat memiliki banyak fungsionalitas dan lengkap untuk membuat program, selain itu alasan lain seperti bahasa Python yang memiliki lisensi *Open Source* dan mudah dikembangkan. Pemahaman tentang *text editor* yang dipakai perlu dilakukan untuk menuliskan kode maka dari itu penggunaan *text editor* Pycharm dipilih.

Pada minggu ketiga, setelah mengenal sekilas tentang bahasa pemrograman Python, tentunya untuk membuat sebuah program dibutuhkan pembelajaran mandiri untuk mengenal lebih jauh konsep dari Python, dan tentunya pembelajaran ini berkaitan dengan fungsi, variabel dan beberapa kode yang diperlukan untuk mencetak hasil kode teks ke layar. Pembahasan selanjutnya mengenai program yang akan dibuat menjadikan sebuah aplikasi yang berfokus pada GUI atau *Graphical User Interface* dengan menggunakan *framework* Tkinter Python dengan bentuk *windows*.

Pada minggu keempat, penulis berpikir untuk berfokus terlebih dahulu terhadap front end dari aplikasi yang akan dibuat, dibutuhkan sebuah tampilan yang menarik, namun simpel untuk dijadikan sebagai rancangan dari *User Interface* form tamu hotel yang menginap. Selain itu, dibuatlah beberapa menu dan komponen *library framework* Tkinter Python untuk menambah fungsionalitas.

Pada minggu kelima, bentuk *form* tamu hotel ini tampak kurang dari segi fungsionalitas dan perkembangan, sehingga diterapkanlah konsep CRUD (*Create, Read, Update, Delete*) ke dalam aplikasi GUI menggunakan *framework* Tkinter ini. Serta menerapkannya ke kodingan yang ada dengan penggunaan beberapa *library* terbaru.

Pada minggu keenam, penulis diharapkan untuk menyimpan data tamu hotel yang telah ada di *form* dengan dihubungkan ke dalam Mysql *database* dimulai dari data nama, alamat, no hp, no ktp dan *email* untuk nantinya ditampilkan kembali jika dibutuhkan.

Pada minggu ketujuh, terjadi masalah yang tidak terduga oleh penulis, seperti beberapa fitur *button* yang menghasilkan *error* ketika dijalankan, menciptakan hasil yang tidak sesuai di layar sehingga perlu adanya pengecekan ulang tentang penempatan variabel, logika pembentukan fungsi dan memperbaiki beberapa *bug*. Selain itu, penulis menambahkan fungsi *search* berdasarkan data tamu hotel mulai dari nama, no hp dan no ktp serta *email* yang berguna untuk melakukan pencarian

data tamu yang telah tersimpan secara spesifik.

Pada minggu kedelapan, penulis berfokus untuk pembuatan fitur edit data yang sudah tersimpan di *database* dengan perubahan secara spesifik ke data tamu tertentu setelah melewati filter *search* data tamu, terdapat *button edit* yang digunakan untuk menampilkan data sebelumnya dari tamu yang nantinya dapat diubah di dalam label dan terdapat *button update* untuk konfirmasi perubahan data baru.

Pada minggu kesembilan, pembuatan aplikasi dilanjutkan dengan sistem basis data transaksi untuk memulai pembuatan *loyalty guest* yang direncanakan sebelumnya, penulis juga mencoba menambahkan beberapa fitur CRUD *create, read, update, delete*.

Pada minggu kesepuluh hingga keduabelas, model dari *database* transaksi yang ada harus diperbaharui agar terlihat lebih menarik, modern dan memperindah dari segi *style, background*, agar setiap data yang masuk pun dapat dengan mudah di bedakan.

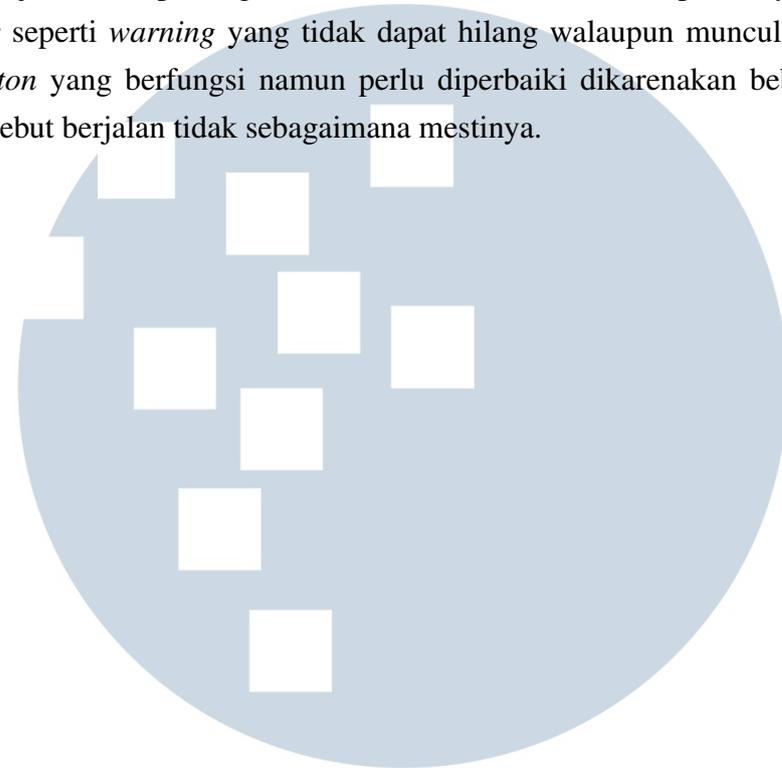
Pada minggu kesebelas hingga ketigabelas, pencarian logika tentang sistem *point loyalty guest* untuk mendapatkan jumlah yang benar dari *benefit* yang bisa didapatkan oleh tamu ketika *point* yang di akumulasi kan dari hasil penginapan sudah mencapai batas tertentu. Adanya variabel poin baru dan lama sebagai acuan, ketika *point* baru didapatkan dari perhitungan faktor jenis kamar, lama menginap, jumlah kamar. Di sisi lainnya, untuk mendapatkan *point* lama penarikan data dari kolom *record* sebelumnya yang telah terisi. Selanjutnya, adanya penambahan fitur untuk memvalidasi tanggal tamu yang menginap mulai dari *check-in* dan *check-out*.

Pada minggu keempat belas, terdapat penambahan sistem *child* dan *parent* untuk menamai *window* agar dengan mudah nantinya di pindah sesuai kemauan.

Pada minggu kelimabelas hingga keenambelas, model *database* transaksi *treeview* yang ada kemudian di koneksi kan dengan *sqlite3* dan menggabungkan aplikasi secara keseluruhan dari *form database* tamu ke sistem *loyalty guest* menggunakan *database* detail transaksi tamu.

Pada minggu ketujuh belas hingga kedelapan belas, penambahan fungsionalitas seperti *button remove-all data, moveup data, move down data, clear entry*. Serta menambahkan fungsionalitas untuk sistem *poin button* untuk mendapatkan total hari tamu menginap yang didapatkan dari *pick calender check-in* dan *check-out*.

Pada minggu kesembilanbelas hingga duapuluh satu, Perbaikan *bug* yang banyak menjadi fokus penting dikarenakan tidak sedikit di tampilan layar terdapat *bug minor* seperti *warning* yang tidak dapat hilang walaupun muncul hasil yang benar, *button* yang berfungsi namun perlu diperbaiki dikarenakan beberapa dari *button* tersebut berjalan tidak sebagaimana mestinya.



UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.1 Proses Pelaksanaan

Perancangan dan pembuatan aplikasi untuk melihat data tamu yang mengingap memerlukan *hardware software* yang cukup untuk membuat program. Berikut merupakan *hardware* dan *software* selama pembuatan aplikasi :

1. *Hardware* : ASUS VIVO BOOK A442U
 - (a) Sistem Operasi : Windows 10 Home Single Language 64bit
 - (b) *Processor* : 10th Gen Intel Core i5 8250
 - (c) RAM : 8 GB DDR4
 - (d) *Graphic Card* : Intel UHD Graphics Family
2. *Software*
 - (a) Pycharm Community Edition 2021.1.3
 - (b) MySQL Workbench 8.0
 - (c) Bahasa pemrograman Python dengan framework Tkinter

3.2.2 System Requirement

Persyaratan dari pekerjaan yang penulis lakukan yang diberikan oleh Pak Adidharma, berikut adalah *requirement* yang dibutuhkan untuk membuat aplikasi:

A. Pembuatan Database

Database profil tamu CRM *Customer Relationship Management* dan penggunaan *database* transaksi riwayat yang menampilkan secara spesifik riwayat satu tamu yang dipilih terpisah.

B. Aplikasi dengan *database* CRM form tamu

Menambahkan data tamu ke dalam *database*, membuat fitur *searching* data tamu hotel yang mengingap dengan beberapa *filter* dari masing-masing data yang dimasukkan untuk mencari dan menampilkan secara spesifik data tamu yang diperlukan. Penambahan fitur *update* untuk mengganti data yang tidak sesuai

di dalam *database*. Dilengkapi dengan fitur *add* yang berguna untuk membuka *window* baru yang memunculkan aplikasi transaksi riwayat untuk menambahkan riwayat penginapan tamu.

C. Aplikasi dengan *database* transaksi riwayat

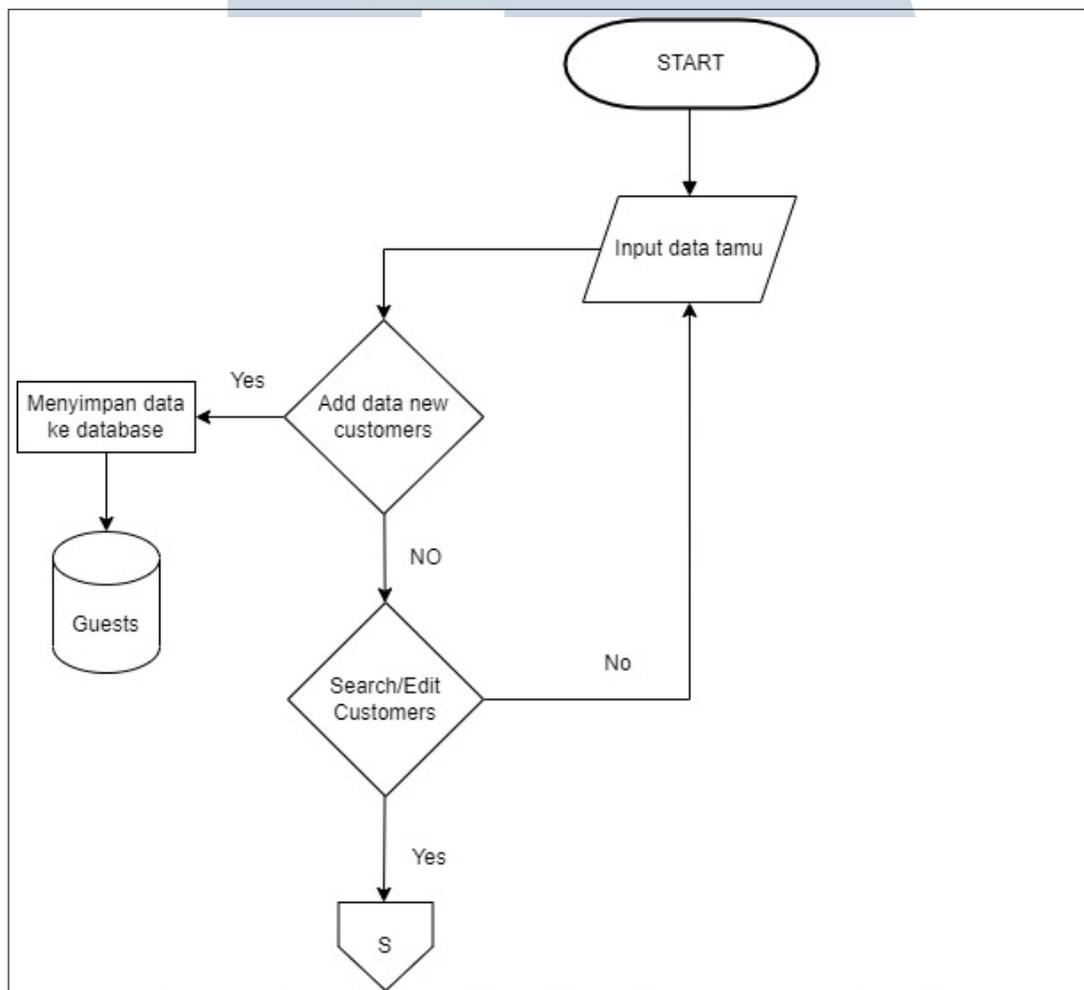
Kalkulasi *point* secara otomatis untuk menampilkan data tamu yang sudah bisa mendapatkan *benefit* dari hotel, penggunaan tipe kamar dilambangkan sebagai angka untuk menentukan *point* yang didapatkan tamu ketika menginap sekali. Kemudian menggunakan *calendar input* untuk mengambil tanggal *check-in* dan *check-out*. Dilanjutkan dengan validasi tanggal *customer* menginap pilihan *check-out* yang harus lebih kecil dari *check-in*. Lalu penggunaan fitur *drop down* untuk mendapatkan poin dari tipe kamar yang dipilih.



3.2.3 Perancangan

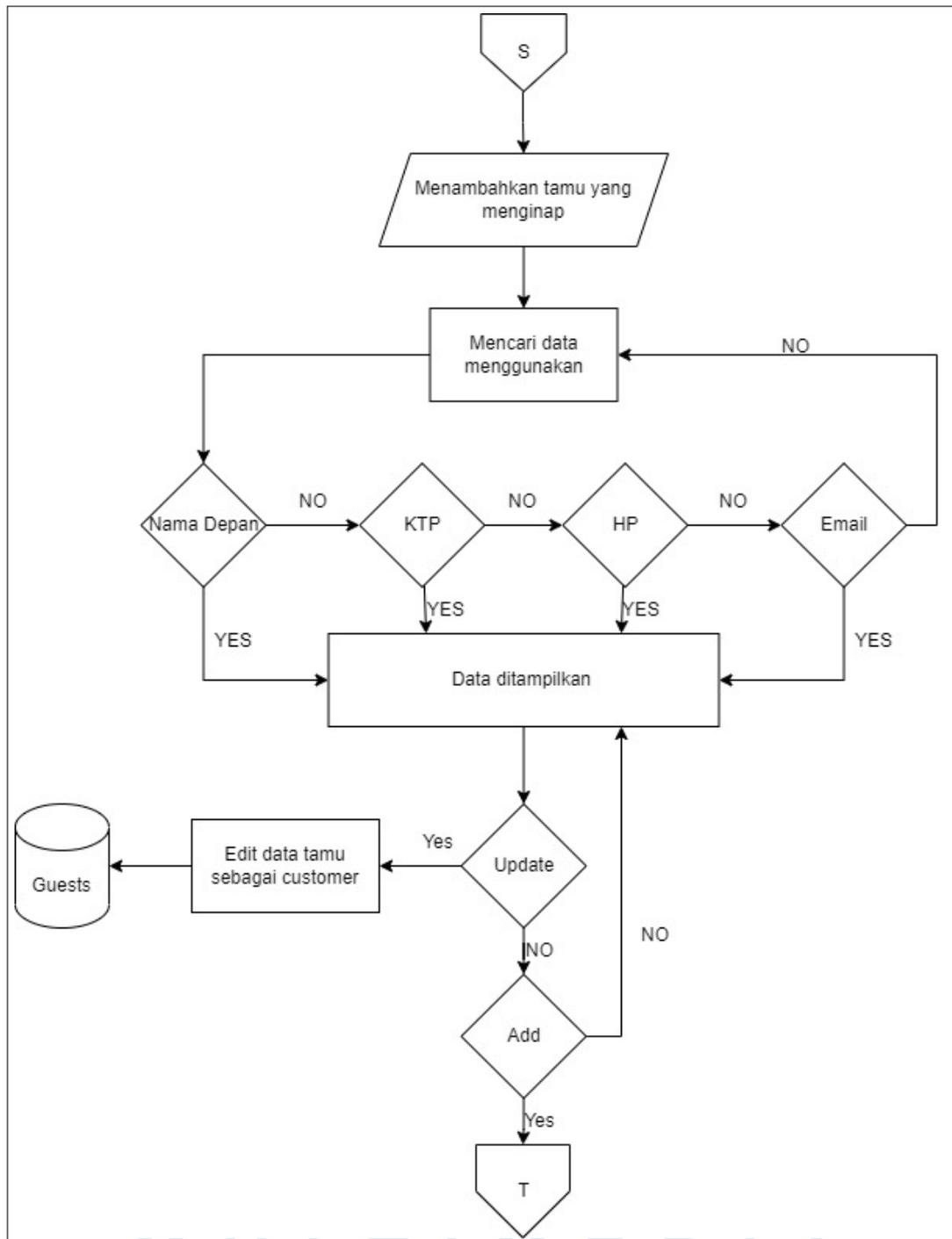
Berikut merupakan gambaran jalannya program aplikasi *loyalty guest* dari satu proses ke proses lainnya untuk memudahkan pemahaman.

A. Skema Flowchart



Gambar 3.1. Flowchart data tamu

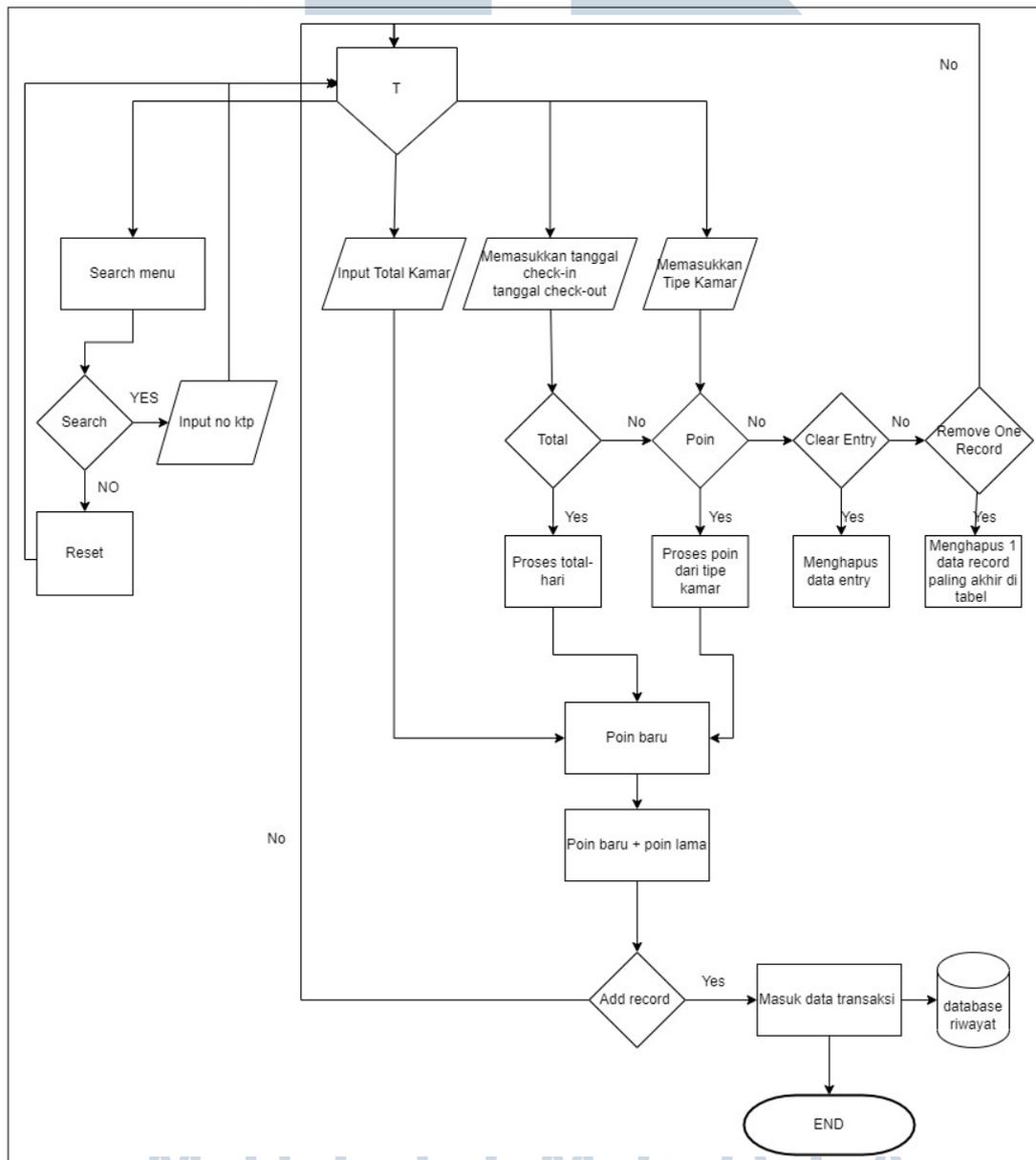
Gambar 3.1 menunjukkan cara untuk memasukkan data pribadi yang penting tamu hotel yang menginap, lalu dilanjutkan ke pembuatan keputusan untuk menambahkan data tamu dan menyimpannya di dalam *database* tabel *guests* atau masuk ke fitur *search and edit customers*.



Gambar 3.2. Flowchart search filter

Gambar 3.2 menunjukkan kelanjutan dari *flowchart* gambar 3.1 dan dimulai dengan penambahan tamu dan kemudian dapat dicari dengan beberapa filter mulai dari nama depan, ktp, hp dan email yang nantinya akan menampilkan data secara spesifik dengan filter yang

telah dipilih. Kemudian dilanjutkan dengan pemilihan *button* antara *update* dan *add*. *Update* untuk mengganti atau menyunting data salah yang sudah tersimpan sebelumnya. *Add* untuk menambahkan transaksi yang dilakukan oleh tamu hotel di *database* yang berbeda.

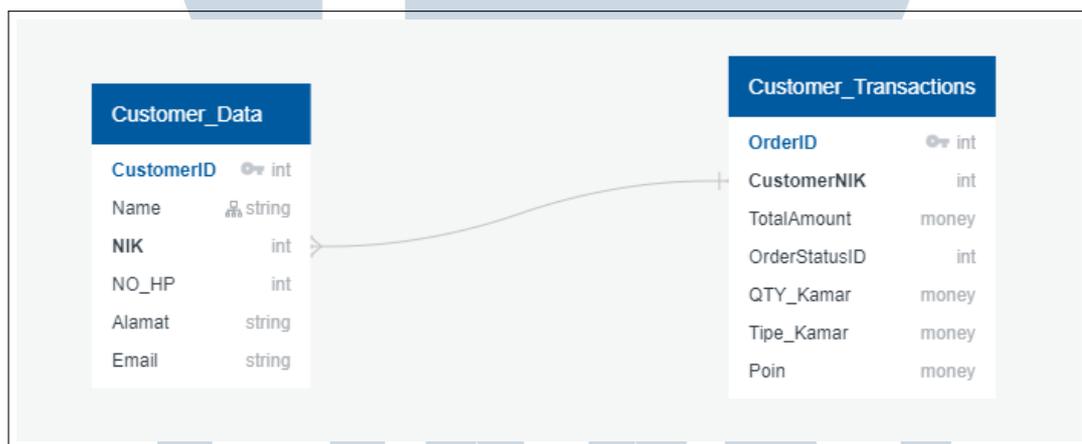


Gambar 3.3. Flowchart riwayat transaksi

Gambar 3.3 menunjukkan kelanjutan dari *flowchart* gambar 3.2 dan dimulai dengan *input* yang dilakukan dari total kamar, tanggal *check-in* dan tanggal *check-out* serta tipe kamar. Nantinya, proses akan memunculkan total-hari dan *point* yang didapatkan

dari tipe kamar. Selain proses tersebut ada proses *clear entry* untuk menghapus *data entry* dan proses *remove one record* untuk menghapus *data record* paling akhir di tabel. Setelah proses-proses tersebut selesai, maka poin baru akan ter kalkulasi dalam sistem dan ditambahkan dengan *point* lama yang merujuk pada transaksi terakhir. Ketika semua *data entry* telah terisi, maka penggunaan *add record* untuk memasukkan data transaksi ke *database* riwayat dapat dilakukan. Selain itu, ada menu *search* untuk menampilkan data transaksi riwayat tamu secara spesifik 1 orang saja dan menu *reset* untuk menampilkan data transaksi riwayat tamu secara keseluruhan.

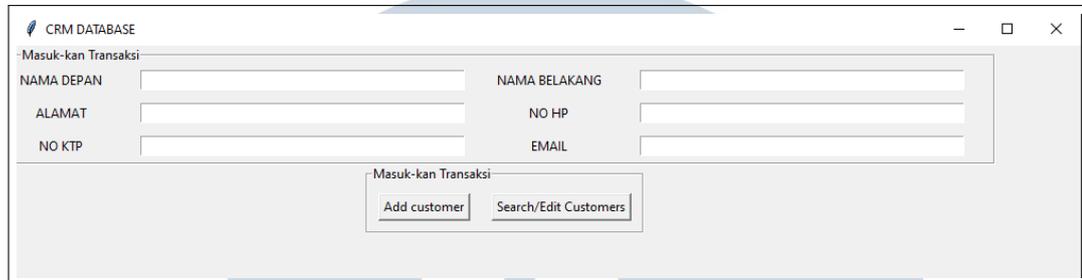
B. Db Schema



Gambar 3.4. Database Skema

Bentuk *database* skema pada Gambar 3.3 terdiri dari dua tabel yang memiliki relasi atribut NIK dari tabel yang ada di gambar skema bawah ini. Ada *database* tamu yang disimpan untuk menjadi referensi dan menjadikan *customer id* sebagai *primary key* dan *database* transaksi yang terdiri dari *attribute point* untuk menampilkan program *loyalty guest reward points* dan menempatkan *order id* sebagai *primary key*.

3.2.4 Implementasi



CRM DATABASE

Masuk-kan Transaksi

NAMA DEPAN NAMA BELAKANG

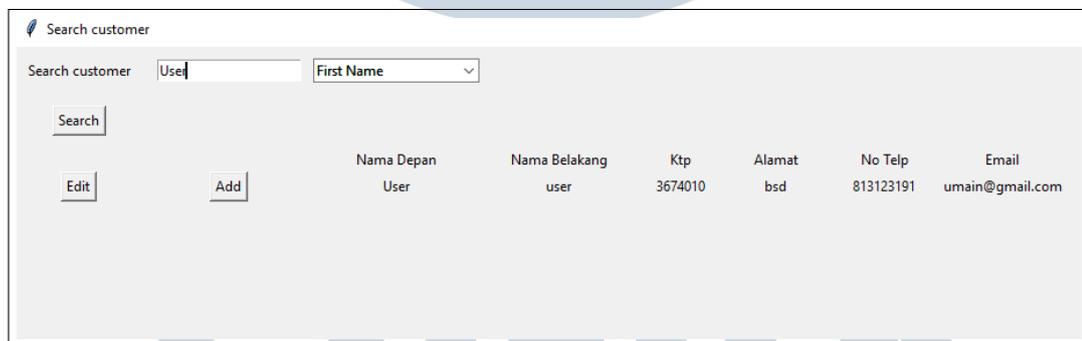
ALAMAT NO HP

NO KTP EMAIL

Masuk-kan Transaksi

Gambar 3.5. Tampilan *window root* awal

Tampilan awal Gambar 3.4 dari *database* CRM yang ditujukan untuk menjadi *database* tamu hotel yang menginap terdiri dari beberapa label seperti nama depan, alamat, no KTP, nama belakang, no hp, hingga *email* dari tamu. Lalu ketika *button add customer* di klik, maka data yang akan disimpan ke *database* Mysql di eksekusi dan di *commit* dengan *values* yang ingin dimasukkan ke tabel *guests*.



Search customer

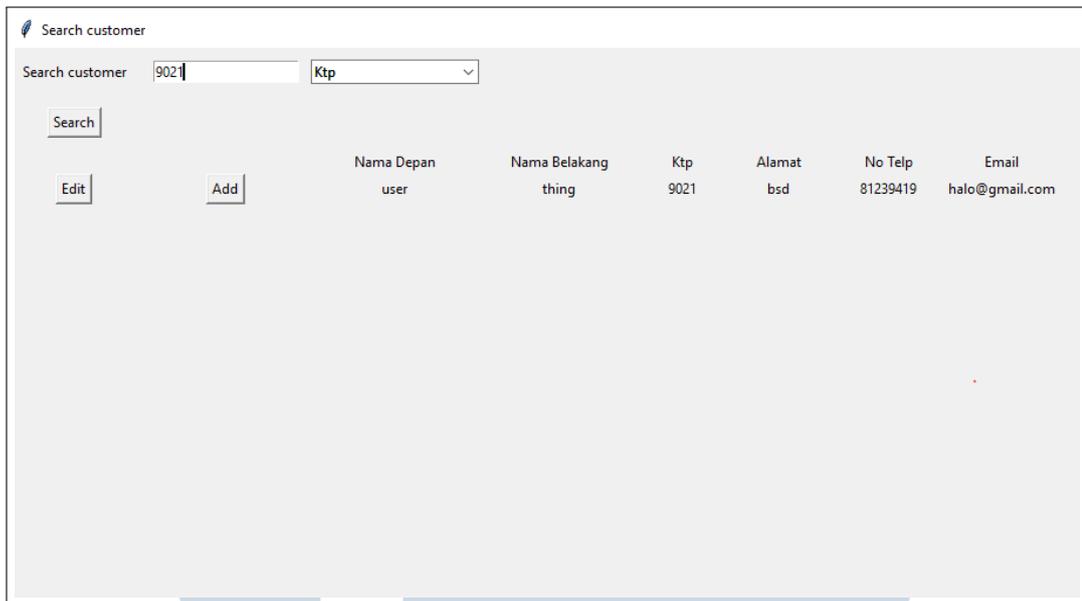
Search customer

Nama Depan	Nama Belakang	Ktp	Alamat	No Telp	Email
User	user	3674010	bsd	813123191	umain@gmail.com

Gambar 3.6. Tampilan pencarian tamu dengan *firstname*

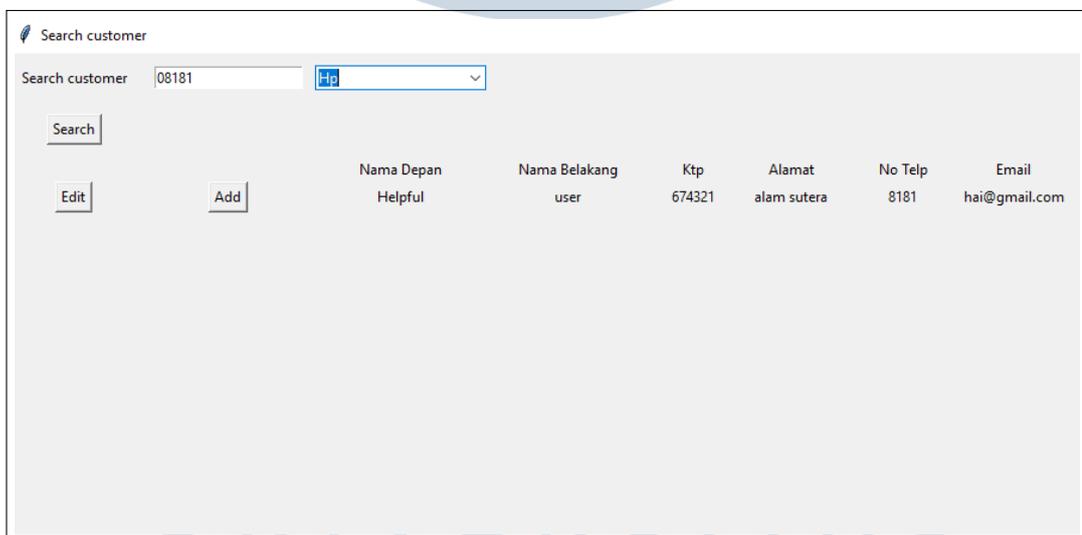
Berikut merupakan tampilan Gambar 3.5 untuk mencari tamu dengan penggunaan fitur *search* dengan pilihan yang ada di *drop down box first name*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



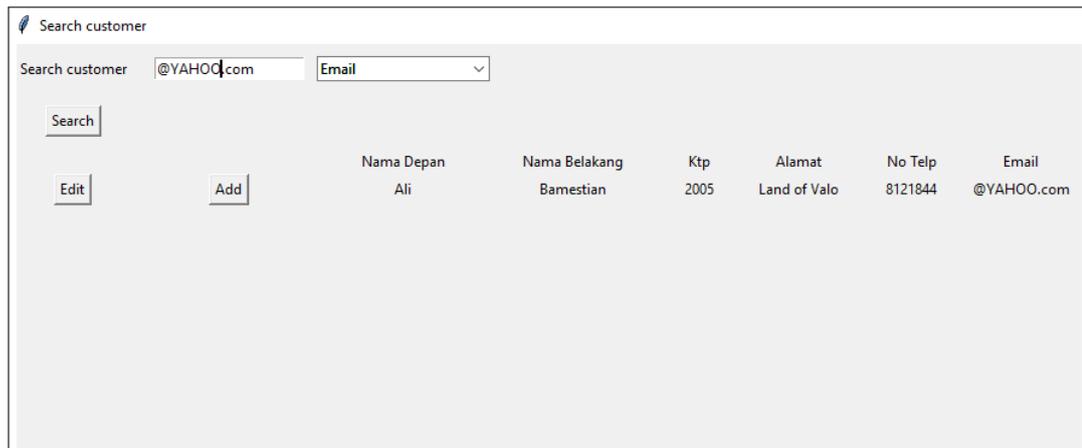
Gambar 3.7. Tampilan pencarian tamu dengan KTP

Berikut merupakan tampilan Gambar 3.6 untuk mencari tamu dengan penggunaan fitur *search* dengan pilihan yang ada di *drop down box* KTP.



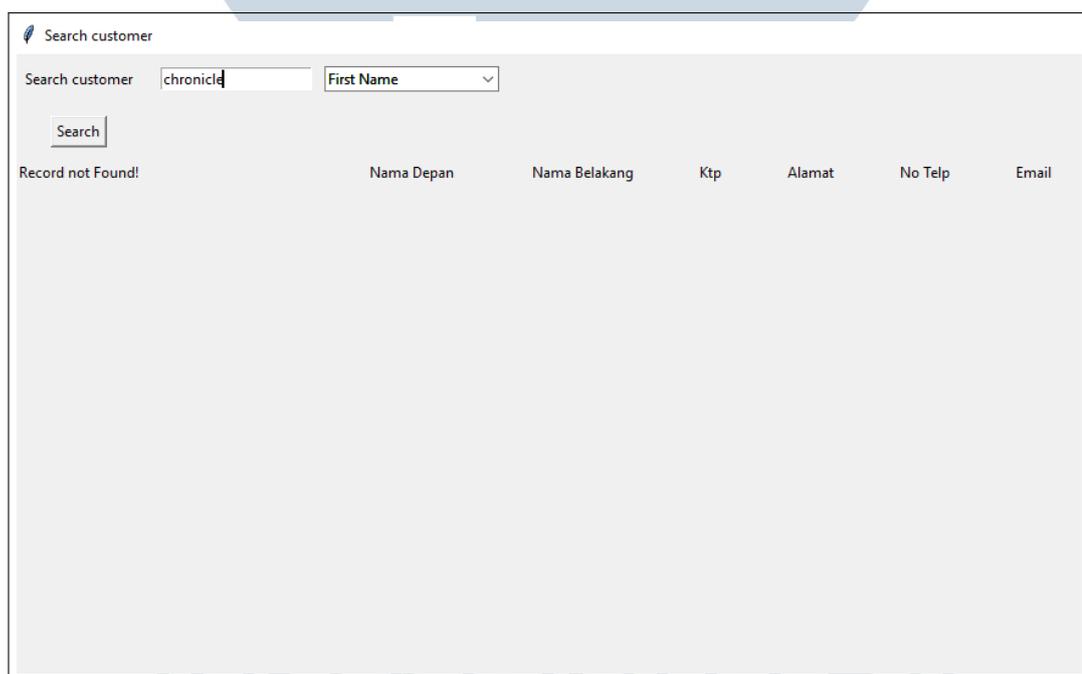
Gambar 3.8. Tampilan pencarian tamu dengan nomor hp

Berikut merupakan tampilan Gambar 3.7 untuk mencari tamu dengan penggunaan fitur *search* dengan pilihan yang ada di *drop down box* Hp.



Gambar 3.9. Tampilan pencarian tamu berdasarkan email

Berikut merupakan tampilan Gambar 3.8 untuk mencari tamu dengan penggunaan fitur *search* dengan pilihan yang ada di *drop down box email*.

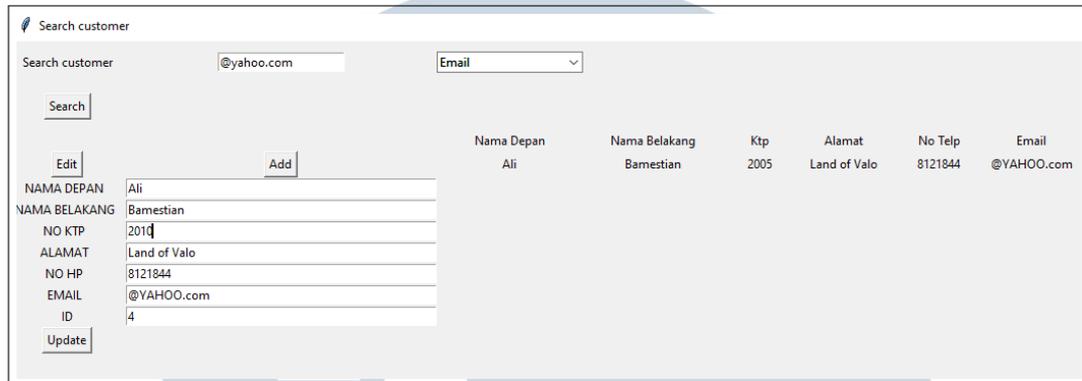


Gambar 3.10. Tampilan data tidak ditemukan

Setelah itu, jika *button search* dijalankan maka pencarian data sebelumnya tidak terdaftar di *database* tamu maka melalui *window* pencarian *customer* ini akan ditampilkan *warning message* data *record* tidak ditemukan di *database* tamu seperti pada Gambar 3.9

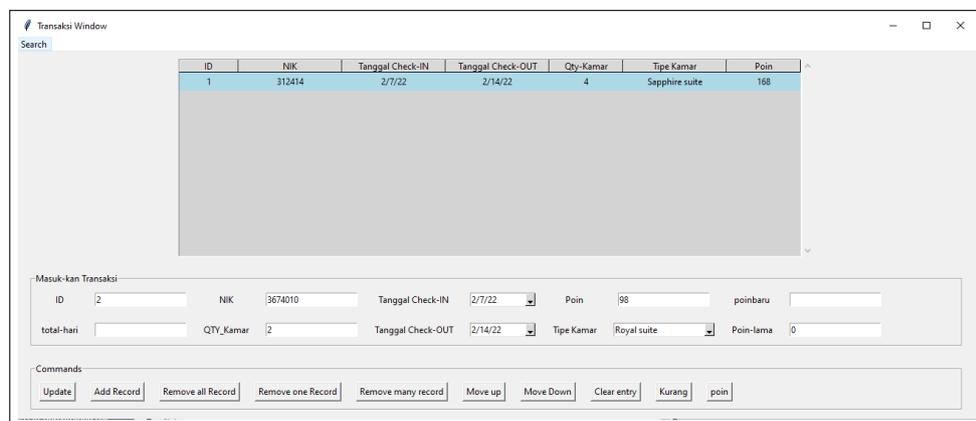
Jika penulis tidak melakukan pemilihan terhadap 1 opsi *drop down*

combo box dari fitur *search* ini, maka *warning* lupa untuk memilih opsi yang ada di *box* tersebut.



Gambar 3.11. Sebelum *update*

Dilanjutkan dengan *button* edit yang terdapat di *window* di Gambar 3.10 tersebut yang digunakan untuk mengganti dan memperbaharui informasi dari tamu yang menginap secara otomatis dengan cara ditimpa dengan mengisi data yang terbaru di variabel *input update record* dengan penggunaan variabel *global* agar dapat digunakan untuk beberapa fungsi.



Gambar 3.12. Tampilan awal aplikasi sistem poin

Dilanjutkan dengan *add button* yang menampilkan riwayat penginapan untuk tamu yang menginap secara individual di transaksi *window* dengan format *treeview* seperti di Gambar 3.11.

Transaksi Window

Search

ID	NIK	Tanggal Check-IN	Tanggal Check-OUT	Qty-Kamar	Tipe Kamar	Poin
1	312414	2/7/22	2/14/22	4	Sapphire suite	168
2	3674010	2/7/22	2/14/22	2	Royal suite	98

Masuk-kan Transaksi

ID NIK Tanggal Check-IN Poin poinbaru

total-hari QTY_Kamar Tanggal Check-OUT Tipe Kamar Poin-lama

Commands

Update Add Record Remove all Record Remove one Record Remove many record Move up Move Down Clear entry Kurang poin

Gambar 3.13. Perhitungan total hari

Selanjutnya mengenai durasi penginapan tamu yang dapat dimasukkan sebagai nilai "total-hari" ke dalam *database* menggunakan fungsi *button* tanggal seperti pada Gambar 3.12.

Poin

Tipe Kamar

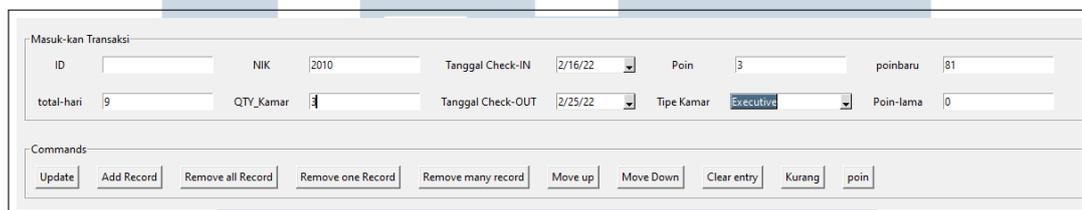
Gambar 3.14. Perhitungan poin dari tipe kamar yang dipilih

Berikut Gambar 3.13 yang menjelaskan tentang tipe kamar yang menggunakan *drop down box*, setiap dari *values* yang ada masuk ke dalam

fungsi *if*, tipe kamar yang diambil akan menyesuaikan dengan *point* yang di dapat ketika di *insert* ke *pointentry dataframe*.

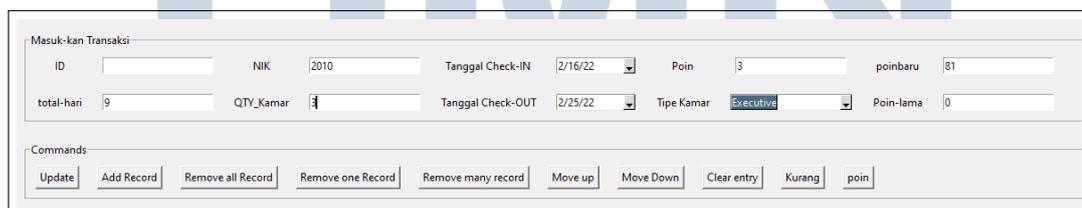


Gambar 3.15. Perhitungan *point* sebelum kamar terpilih



Gambar 3.16. Perhitungan *point* setelah kamar terpilih

Berlanjut pada perhitungan logika dan matematika dari *point loyalty guest* yang berdasarkan pada fungsi jumlah perkalian seperti di Gambar 3.14 yang terdiri dari *qty(quantity)*, *point* kamar yang didapatkan dari pemilihan jenis kamar dan lama menginap menggunakan total-hari. Nantinya di gambar 3.15 *point* yang ada di label setelah diakumulasikan secara otomatis dengan perhitungan yang sudah ada akan menjadi *point* yang dimasukkan ke *database*.



Gambar 3.17. Fitur *button add*

Selanjutnya Gambar 3.16 yang memperlihatkan fitur *add button* yang digunakan untuk menyimpan data ke tabel transaksi dengan iterasi ID mengikuti data yang terakhir dimasukkan sehingga penempatan tidak bertabrakan dengan data lainnya, di sini fitur *add* dapat berfungsi secara maksimal jika semua *entry box* dapat terisi.

3.3 Kendala dan Solusi yang Ditemukan

Selama melakukan pekerjaan magang di Sapphire Sky Hotel BSD selama kurang lebih 5 bulan pasti ada kendala dan solusi yang ditemukan oleh Yustinus Lionardy

3.3.1 Kendala yang ditemukan

Berikut merupakan kendala yang dijumpai selama magang di Sapphire Sky Hotel BSD.

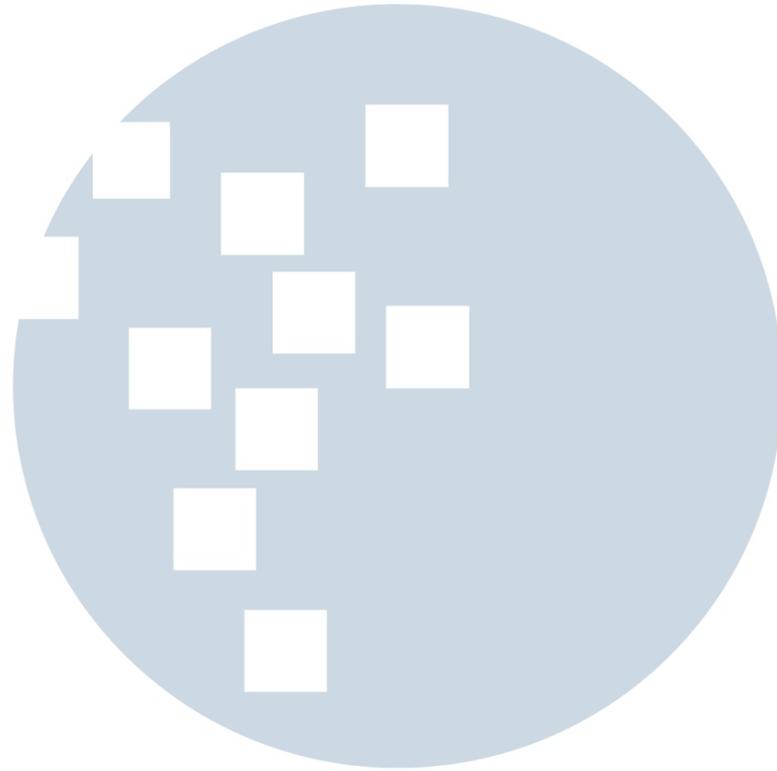
1. Kendala yang penulis temukan berupa *bug* yang muncul dan adanya *warning* walaupun hasil yang benar sudah ada di layar.
2. Kesulitan dalam menerapkan kalkulasi sistem *point* dan menerapkan beberapa kriteria ke dalam kodingan ke *model tree-view*. Sistem *point* yang berkaitan dengan tabel tidak dapat menambahkan kalkulasi *point* transaksi sebelumnya.
3. Kendala lainnya yaitu terkait konektor untuk Mysql. Masalah instalasi *mysql-connector-python* ini memerlukan percobaan tergantung pada kecocokan mysql dengan versi Python yang digunakan.

3.3.2 Solusi yang ditemukan

Berikut solusi yang penulis jumpai selama magang di Sapphire Sky Hotel BSD

1. Kendala ini harus penulis coba perbaiki dengan cara *trial and error* untuk menemukan titik kodingan yang salah, mencoba mengganti beberapa fungsi yang tidak jalan dan menambahkan fitur fungsi baru.
2. Menerapkan logika seperti *reset*, penggunaan variabel baru *point* lama dan *point* baru untuk perhitungan *point* total yang didapatkan.
3. Mencoba tiga cara melakukan instalasi Mysql *connector* yang tidak bekerja namun penulis mencoba *reinstall connector* untuk mencoba kembali walaupun lagi tidak berhasil, lalu akhirnya *uninstall* Mysql dan memindahkan data penyimpanan-nya ke

tempat *storage* lainnya dan *connector* akhirnya dapat berjalan setelah di *install* kembali.



UMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA