LAPORAN FINAL PROJECT PERANCANGAN APLIKASI DESKTOP PERHOTELAN

Mata Kuliah IS431 – Visual Programming

Dosen Pengampu: Agus Sulaiman, S.Kom, M.M

Disusun oleh Kelompok 5 :

Nathan Villbert Kosasih (NIM: 00000069903)

Ongko Fillberto (NIM: 00000070324)

Fernando Yulius Hadiarto (NIM: 00000070570)

Hans Samuel Gunawan (NIM: 00000069882)

Felix Samuel Leo (NIM: 00000070094)

PROGRAM STUDI SISTEM INFORMASI FAKULTAS TEKNIK DAN INFORMATIKA

UNIVERSITAS MULTIMEDIA NUSANTARA

TANGERANG 2023

DAFTAR ISI

BAB I 3
PENDAHULUAN 3
1.1 Latar Belakang 3
1.2 Tujuan 4
1.3 Manfaat 4
BAB II 5
HASIL 5
2.1 Bagaimana Cara Kerja Proses Bisnis? 5
2.2 Fungsi Check-in dan Check-Out 6
2.3 Cara Kerja Aplikasi Desktop 6
2.4 Flowchart Proses Bisnis 8
2.5 Entity Relationship Diagram 9
2.6 DEMO PROGRAM APLIKASI DESKTOP PERHOTELAN 9
2.7 Application Coding 22
BAB III 45
KESIMPULAN 45
3.1 Kesimpulan 45

DESKRIPSI APLIKASI

Dalam perhotelan, memiliki sistem aplikasi yang mempunyai fitur-fitur yang sudah terintegrasi sehingga dapat membantu manajemen perhotelan dalam kepentingan pengurusan hotel. Sistem dalam aplikasi ini dapat mengatur berbagai aspek operasional hotel. Dengan sistem ini dapat meningkatkan efisiensi dalam operasional dan menyediakan layanan tambahan untuk tamu hotel. Secara keseluruhan, aplikasi desktop perhotelan menjadi solusi yang bisa bermanfaat bagi staf-staf dan manajemen hotel dalam perkembangan sistem operasional, kepuasan tamu, dan analisis data untuk dijadikan bahan evaluasi hotel untuk kedepannya.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem operasional perhotelan menjadi elemen yang sangat penting dalam industri perhotelan modern. Seluruh hotel pastinya membutuhkan sistem manajemen yang efektif dan efisien untuk melayani tamu mereka, dengan berkembangnya teknologi yang sangat pesat dan untuk menghadapi persaingan yang semakin ketat maka hotel-hotel sekarang ini harus mengintegrasikan teknologi ke dalam sistem manajemen mereka. Beberapa contoh dari penggunaan teknologi informasi dalam sistem perhotelan yaitu seperti sistem Check-in dan Check-out, pengelola keuangan hotel, manajemen pelayanan tamu, dan lain-lain. Penggunaan IT dalam industri perhotelan ini dapat mempersingkat waktu, meringankan pekerjaan yang manual hingga menggantikan pekerjaan manusia.

Selain manfaat langsung, penggunaan aplikasi perhotelan dapat memberikan manfaat jangka panjang dalam pengambilan keputusan perusahaan dan juga mempermudahkan tamu. Misalnya, hotel dapat lebih memprioritaskan atau memfokuskan pada pelayanan tamu yang lebih personal dan penting yang menciptakan hubungan yang lebih baik lagi antara staff hotel dan tamu dan tamu juga dapat terbantu oleh aplikasinya jika ingin reservasi kamar, menghindari antrian secara langsung dan segala macamnya dapat dilakukan lewat aplikasi. Pada laporan ini akan membahas mengenai projek ini, akan dirancang sistem aplikasi perhotelan yang terintegrasi untuk

memenuhi kebutuhan spesifik hotel. Aplikasi ini dapat mengoptimalkan operasional hotel seperti yang sudah dijelaskan sebelumnya.

1.2 Tujuan

Berikut adalah tujuan dari aplikasi website perhotelan kami :

- Mengefisienkan staf hotel dalam mengelola reservasi seperti data tamu, ketersediaan kamar, check-in dan check-out , dan konfirmasi reservasi pada tamu.
- Memudahkan staf hotel dalam mengelola keuangan akuntansi, contohnya seperti pembayaran tamu atau pemasukan hotel.
- Memudahkan staf hotel dalam mendata informasi tamu, dan riwayat kunjungan.
- Meningkatkan keamanan dalam pengelolaan data keuangan dan data berharga lainnya.

1.3 Manfaat

Berikut adalah manfaat yang dapat diberikan dari aplikasi perhotelan kami:

- Peningkatan produktivitas dan efisiensi operasional hotel

  Menyediakan banyak fitur yang dapat dihandalkan dan mempermudah hotel untuk menjalani semua proses operasional dengan lebih mudah, cepat, dan mengurangkan penggunaan tenaga manusia.
- Peningkatan layanan kepada tamu

  Dengan adanya reservasi yang mudah dan cepat, check-in dan check-out dari aplikasi, informasi mengenai hotel dan semua hal yang berkaitan dengan hotel dapat lebih lagi dalam peningkatan kualitas pelayanan.
- Pengelola keuangan yang lebih mudah

  Karena adanya fitur pengelola keuangan yang sudah terintegrasi, maka pihak hotel dapat lebih mudah untuk mengatur pendapatan dan pengeluaran dan dapat membuat keputusan finansial yang baik.

BAB II

HASIL

Heaven’s Door Hotel Receptionist System

Gambar. *Logo dari aplikasi website 'Heaven's Door Hotel'*

2.1 Bagaimana Cara Kerja Proses Bisnis?

Ketika ada tamu yang berkunjung ke sebuah hotel maka tamu tersebut pastinya akan berhubungan dengan resepsionis hotel tersebut. Aplikasi yang ingin dirancang ini adalah aplikasi desktop yang khusus dibuat untuk memudahkan pekerjaan resepsionis dalam mencatat dan melihat informasi mengenai hotel tersebut mulai dari transaksi pembayaran yang dilakukan para tamu, nomor kamar yang masih kosong, waktu saat tamu melakukan check-in atau check-out, dan informasi tamu yang melakukan kunjungan ke hotel berupa identitas tamu tersebut. Ketika tamu melakukan registrasi pada hotel, resepsionis hanya perlu memasukkan data tamu tersebut pada aplikasi dan resepsionis akan secara mudah memilihkan kamar kosong untuk ditempati bagi pengunjung tersebut. Setelah itu, resepsionis juga akan mencatat jam dimana pengunjung melakukan check-in sehingga waktu penginapan bisa dipantau oleh resepsionis agar pihak hotel dapat mengetahui kapan

batas check-out bagi pengunjung agar housekeeping pada hotel bisa melakukan pekerjaannya tepat waktu. Selain untuk check-in dan check-out, perbedaan tipe kamar pada hotel juga tersedia pada aplikasi sehingga dapat membantu resepsionis untuk memberikan informasi pada setiap tipe kamar terutama jika ada perubahan pada harga kamar sewaktu-waktu. Pembatalan reservasi juga dapat langsung dikelola oleh admin atau resepsionis pada aplikasi yang nantinya akan memberikan kemudahan dalam menerima pemesanan kamar lainnya.

2.2 Fungsi Check-in dan Check-Out

Check-in merupakan proses transaksi dimana tamu melakukan pembayaran pada pihak pengelola hotel untuk menyewa kamar bersama dengan fasilitas-fasilitas yang ada. Sebaliknya check-out adalah proses dimana tamu melapor kepada resepsionis saat tamu sudah selesai dalam melakukan penyewaan kamar lagi. Fungsi dari aktivitas check-in dan check-out itu sendiri untuk memberikan informasi yang cepat dan akurat kepada pekerja hotel bagian resepsionis dalam mendata/memasukan pengunjung ke pilihan kamar yang kosong. Dengan adanya aplikasi seperti ini, proses check-in dan check-out yang dilakukan oleh pengunjung dapat terlampir dengan jelas informasinya sehingga pembukuan hotel lebih tertata dengan rapih. Selain itu, pihak hotel dapat mengetahui waktu tenggat bagi tamu untuk melakukan check-out, jika tidak maka akan ada konsekuensi jika waktu check-out tamu melebihi batasnya.

2.3 Cara Kerja Aplikasi Desktop

Pada tahap awal saat pengunjung ingin memesan kamar hotel sebagai tempat penginapan, Resepsionis mendata pengguna yang ingin melakukan pemesanan kamar dengan mencantumkan data identitas pengunjung seperti nama, jenis kelamin, nomor hp, nomor ktp, dan lainnya. Lalu resepsionis akan mendaftarkan dan memilih kamar kosong untuk ditempati oleh tamu sesuai dengan tipe kamar yang mereka pilih. Kemudian setelah melakukan pembayaran, resepsionis akan meng-input transaksi tersebut ke dalam aplikasi sehingga tercatat dalam laporan transaksi. Pelanggan bisa melakukan pembayaran dengan metode pembayaran yang disediakan dalam sistem aplikasi seperti mobile banking, cash,

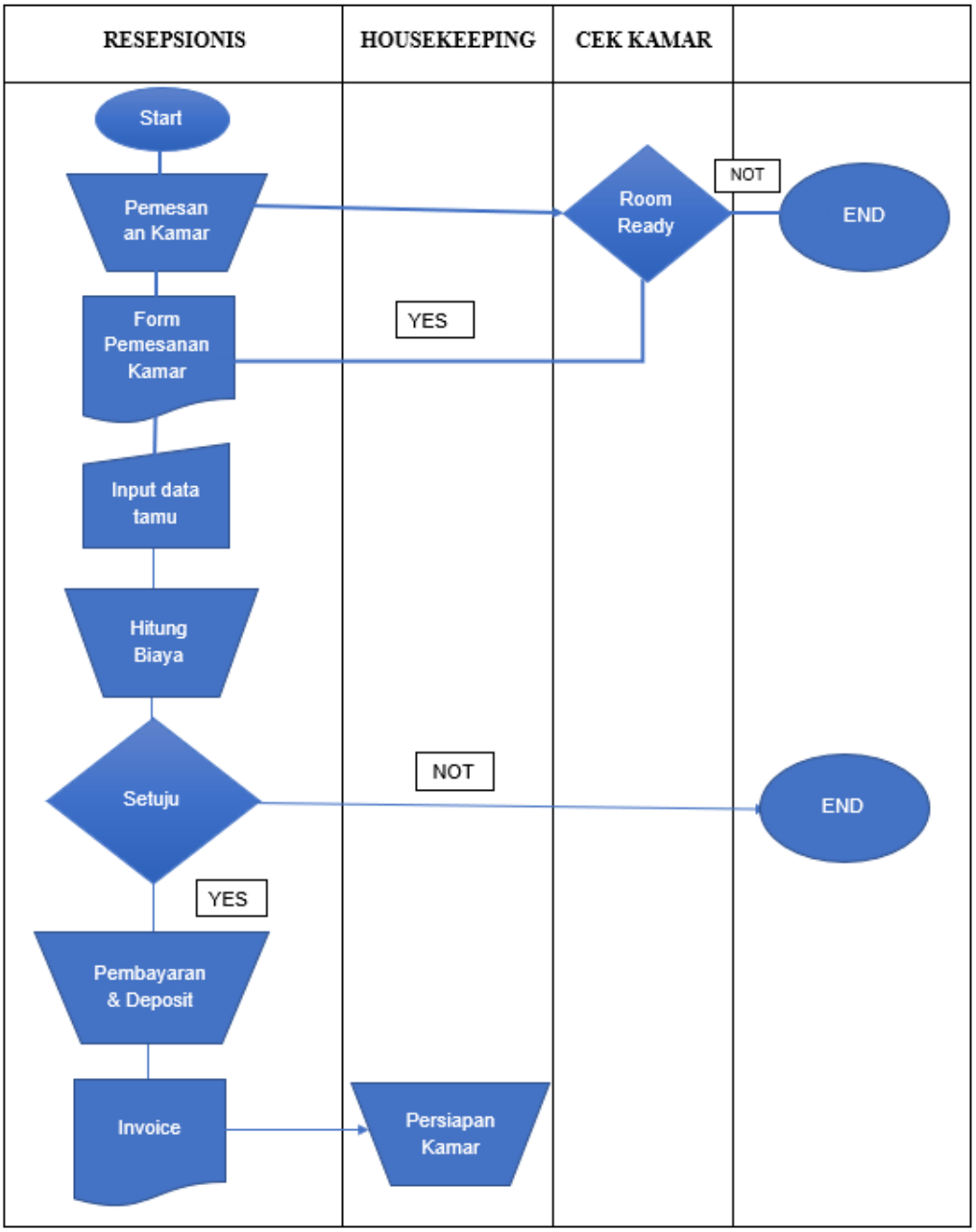
debit, maupun kredit. Lanjut pada tahap check-in, pengunjung dapat menggunakan kamarnya, yang nantinya akan masuk dalam data manajemen hotel kamar mana yang sudah terisi dan kamar kosong yang siap untuk di check-in maupun kamar kosong yang baru selesai di check-out. Dalam aplikasi desktop, akan tersedia informasi batas waktu bagi tamu untuk melakukan check-out dan bila tamu melakukan check-out melebihi batas tersebut, maka deposit yang dibayar oleh tamu diawal akan ditarik. Saat check-out, aplikasi juga akan menginformasikan jika tamu melakukan penambahan biaya seperti extra bed, pemesanan makanan, dan lainnya. Menu aplikasi desktop tersebut akan tampil sebagai berikut:

Menu Pada Aplikasi Desktop :

- Log-in : sebelum masuk ke pilihan menu, pihak resepsionis akan diminta untuk log-in terlebih dahulu agar dapat mengakses ke tahap selanjutnya.
- Reservasi on-site: salah satu fitur di menu ialah registrasi pengunjung yang belum melakukan reservasi kamar, dimana akan memasukkan data data dari tamu berupa nama pengunjung, jenis kelamin, no hp dan lain-lain. Setelah itu akan lanjut ke pemilihan kamar kemudian ke tahap pembayaran.
- Pre-served room : fitur ini untuk pengunjung yang telah melakukan reservasi kamar sebelumnya sehingga resepsionis dapat mengetahui jika pengunjung sudah melakukan reservasi atau belum sehingga dapat lanjut ke tahap check-in.
- Room : fitur ini untuk membantu resepsionis mengetahui jumlah ketersediaan kamar sesuai tipe tipe kamar tersebut beserta harga kamarnya pada saat itu.
- Check-out: menu ini untuk mengakhiri masa penginapan pengunjung setelah itu mengeluarkan tagihan biaya biasa tambah dengan biaya tambahan seperti extra bed, pemesanan makanan, penggunaan fasilitas lainnya dan ada denda ketika pengunjung melebihi batas waktu check-out.
- Laporan : Menu ini berguna untuk melakukan set laporan yang akan dicatat sebagai laporan statistik hotel oleh pihak manajemen hotel setiap harinya, sehingga dapat dijadikan bahan evaluasi hotel untuk perencanaan dan pemasaran kedepannya.

- Keuangan :Menu ini digunakan untuk menampilkan laporan keuangan hotel seperti transaksi pembayaran. Laporan dapat diatur dalam periode tertentu, seperti harian, mingguan, atau bulanan.

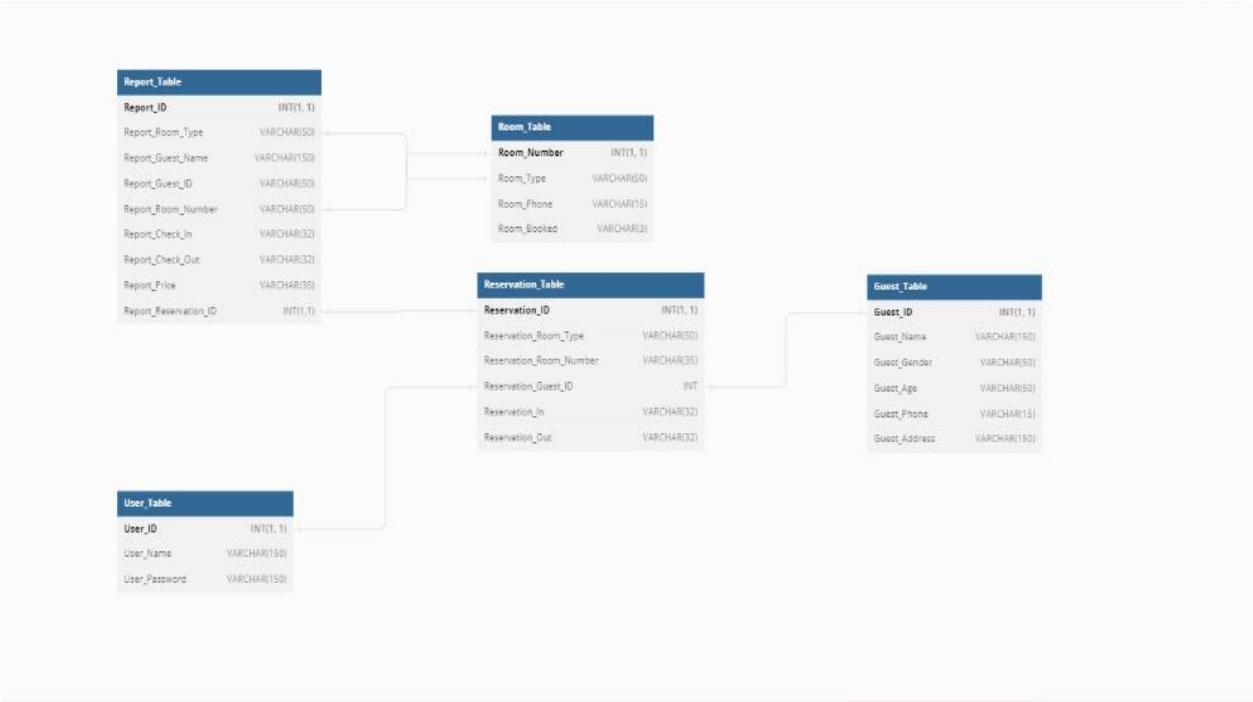
2.4 Flowchart Proses Bisnis



Gambar diatas merupakan flowchart alur dari penggunaan aplikasi rancangan kami. Setelah staff / resepsionis melakukan login kedalam sistem, mereka bisa melakukan

pemesanan kamar, apabila tidak tersedia kamar maka akan langsung berakhir, tetapi jika ada amar yang tersedia maka resepsionis akan mengisi form pemesanan kamar. Setelah pengisian form pemesanan kamar, resepsionis akan input data tamu kemudian hitung biaya dan jika sudah dikonfirmasi dengan pengunjung maka akan dilanjutkan ke tahap pembayaran & deposit,setelah itu diberikan invoice dan kemudian kamar yang dipesan akan dipersiapkan oleh staff hotel untuk tamu tersebut, namun jika pengunjung hotel tidak bersedia / setuju dengan harga kamar maka akan langsung berakhir.

2.5 Entity Relationship Diagram



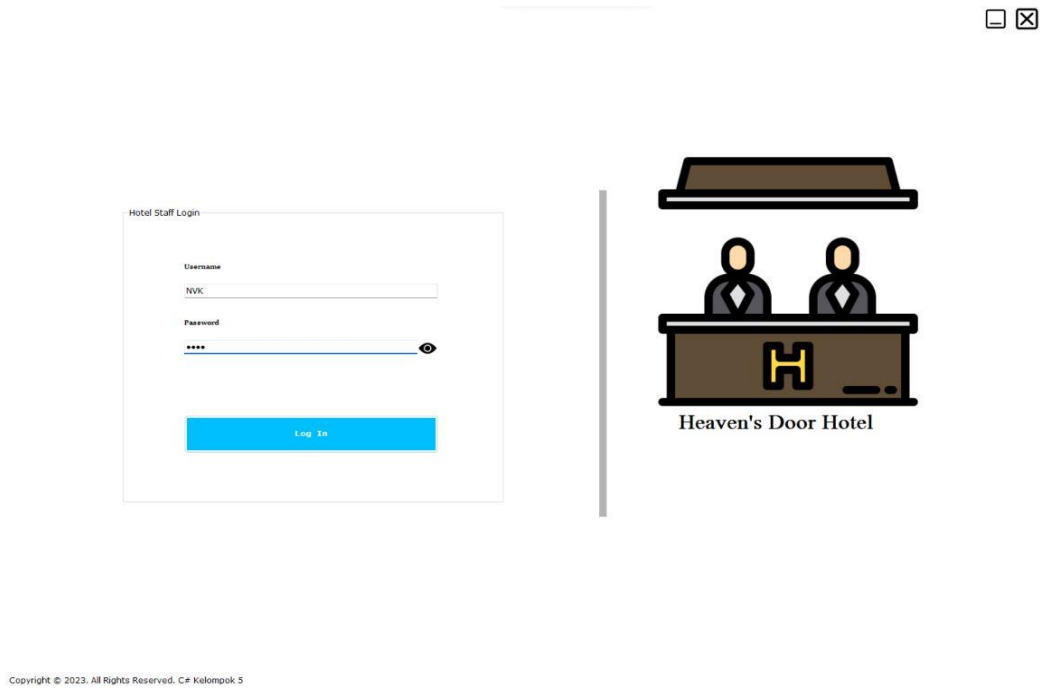
Gambar. *Entity Relationship Diagram*

2.6 DEMO PROGRAM APLIKASI DESKTOP PERHOTELAN

1 . Login Page

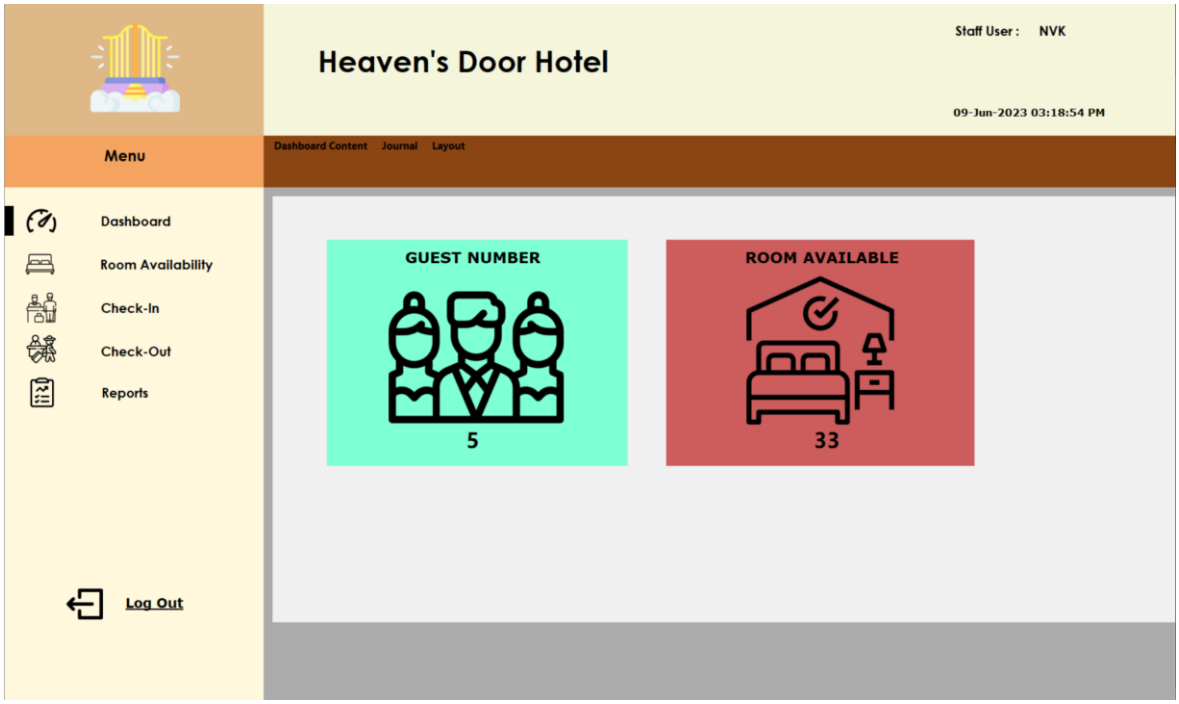
Pada tampilan login ini berguna untuk staff hotel yang akan melakukan login ke dashboard data hotel. Pada page ini staff hotel akan mengisi username dan password mereka. Pada

bagian password staff hotel dapat menggunakan mode password tidak terlihat. Tombol log in berguna untuk lanjut ke halaman dashboard.



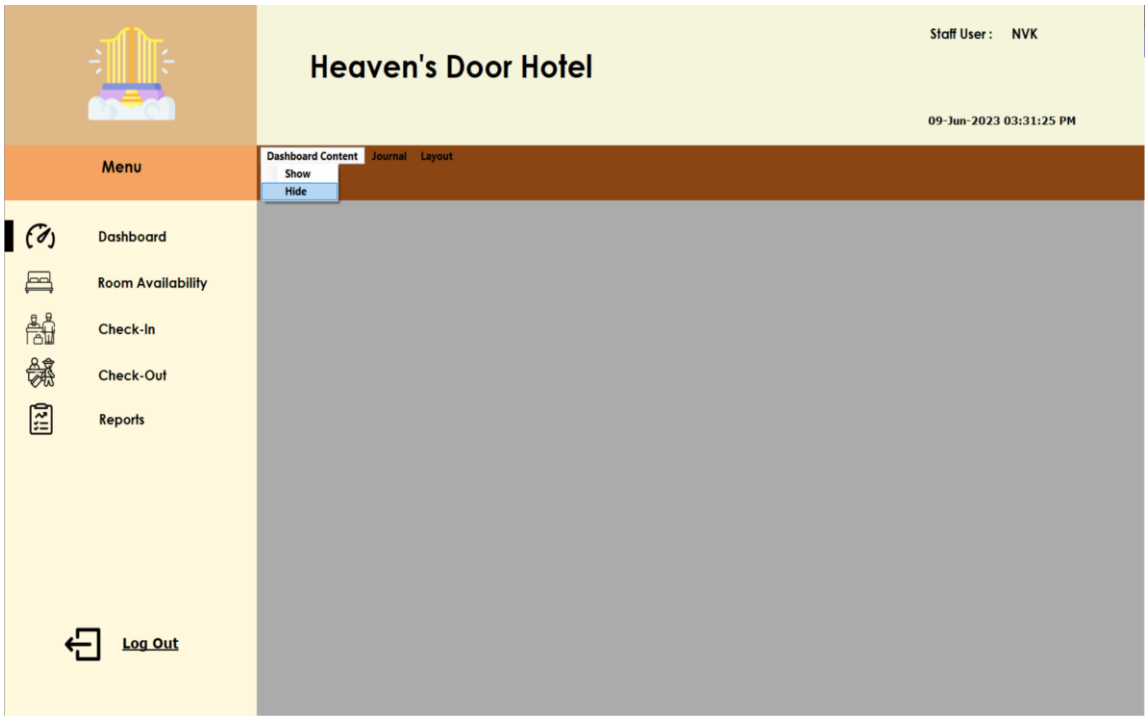
2. Dashboard Aplikasi

Pada halaman dashboard merupakan tampilan utama dari aplikasi desktop *Heaven's Door Hotel* yang memiliki tampilan mengenai *Guest Number* untuk menampilkan data tamu yang ada, sedangkan *Room Available* untuk menampilkan jumlah ruangan atau kamar hotel yang tersedia. Di ujung kanan atas layar terdapat nama dari staff hotel serta waktu nyata yang bergerak seiring aplikasi dibuka. Lalu dibawah *banner* hotel terdapat 3 menu strip yang masing-masing menunya adalah pilihan untuk menunjukkan *dashboard content*, *Journal* yang berisi MDI form *memopad*, *guest wishlist*, dan *room detail*, serta pilihan setting *layout.*



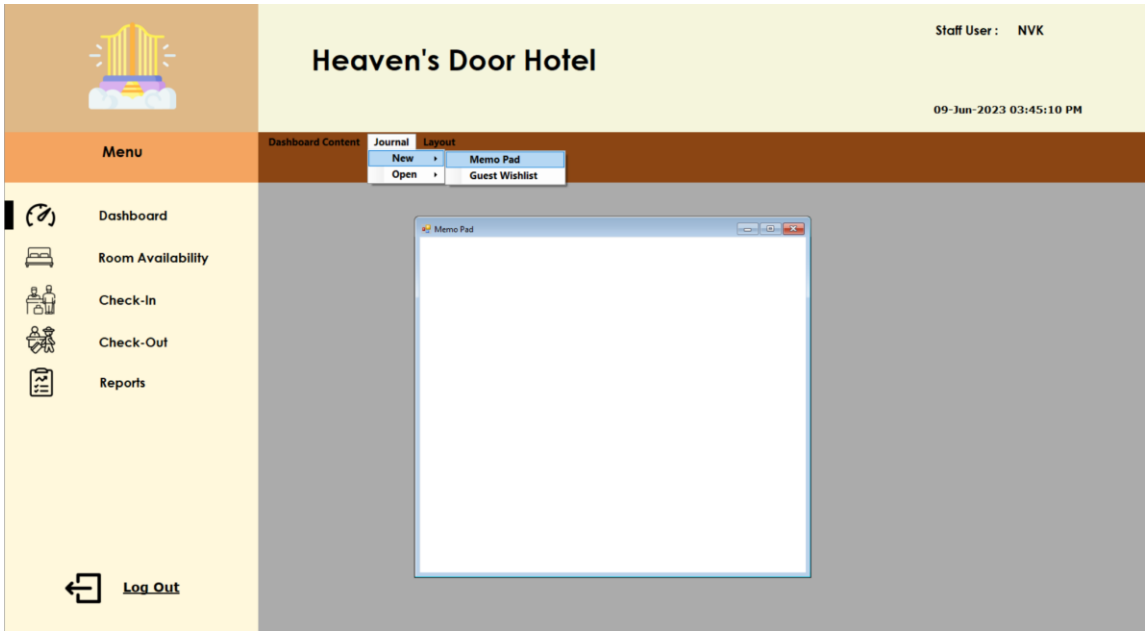
3. Menu Strip - Dashboard Content

Menu strip pertama pada tampilan dashboard adalah *dasboard content* yang memberikan pilihan apakah konten gambaran *guest number* & *room available* ingin ditampilkan atau tidak.

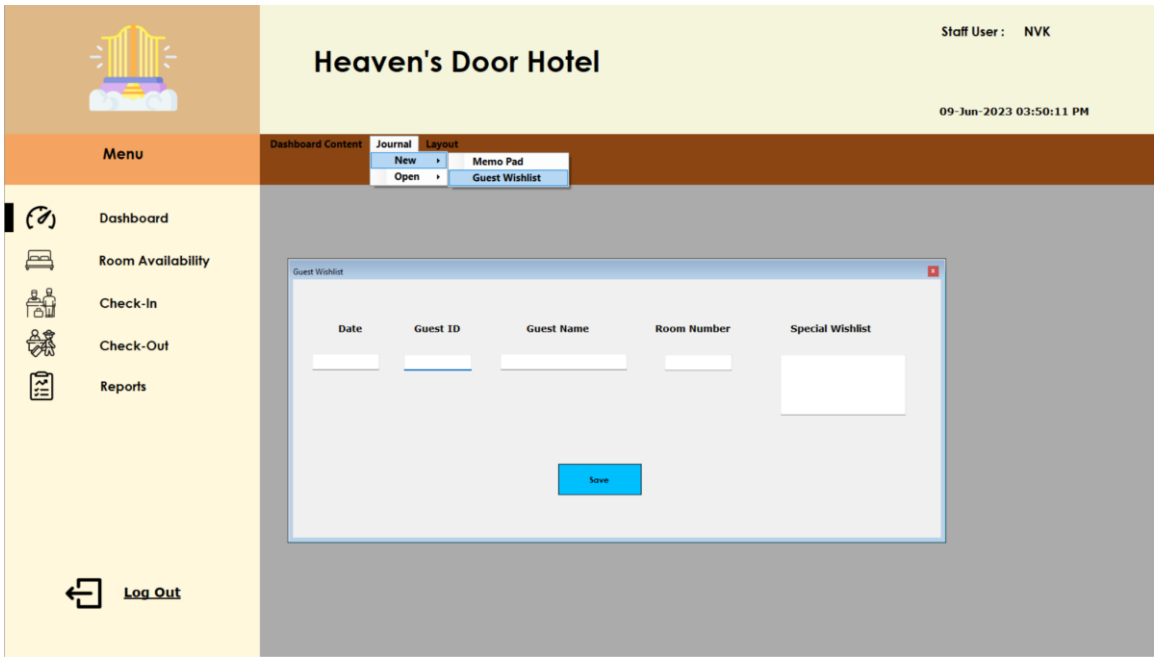


4. Menu Strip - Journal

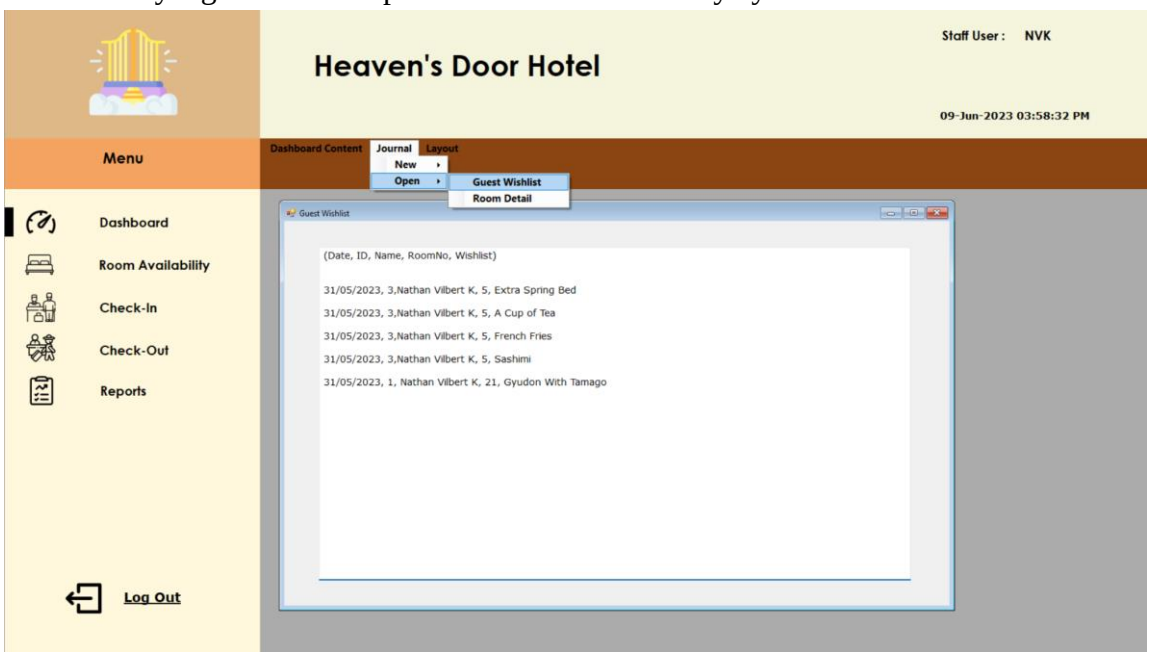
Pada menu strip *journal* terdapat 2 pilihan yaitu *new* & *open* dimana pilihan *new* akan berisi pilihan *memo pad* dan *guest wishlist. Memo pad* berfungsi untuk menampilkan MDI Form yang berbentuk seperti *notepad* atau catatan yang memberikan resepsionis kemudahan untuk mencatat hal-hal kecil yang penting.



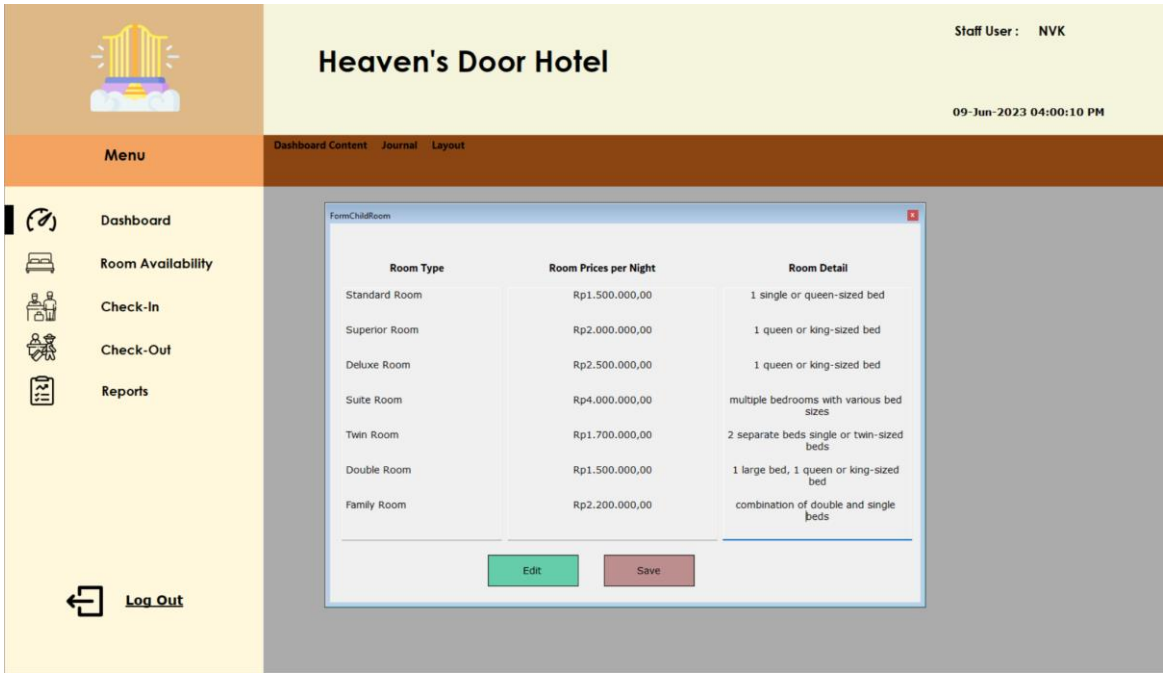
Guest wishlist juga memberikan metode yang sama tetapi menu *guest wishlist* memberikan pencatatan khusus terhadap tamu hotel yang mencakup data tamu seperti tanggal, ID, nama tamu, nomor kamar, serta keinginan khusus tamu. Berbeda dengan *memopad*, MDI Form pada submenu *guest wishlist* memiliki metode *file and stream* dimana data yang dicatat oleh resepsionis akan di update ke dalam file txt bernama Guest Wishlist.txt.



Dibawah menu *new* terdapat menu *open* yang berguna untuk membuka dan membaca file txt dari file yang telah dibuat pada menu *new* sebelumnya yaitu Guest Wishlist.txt.

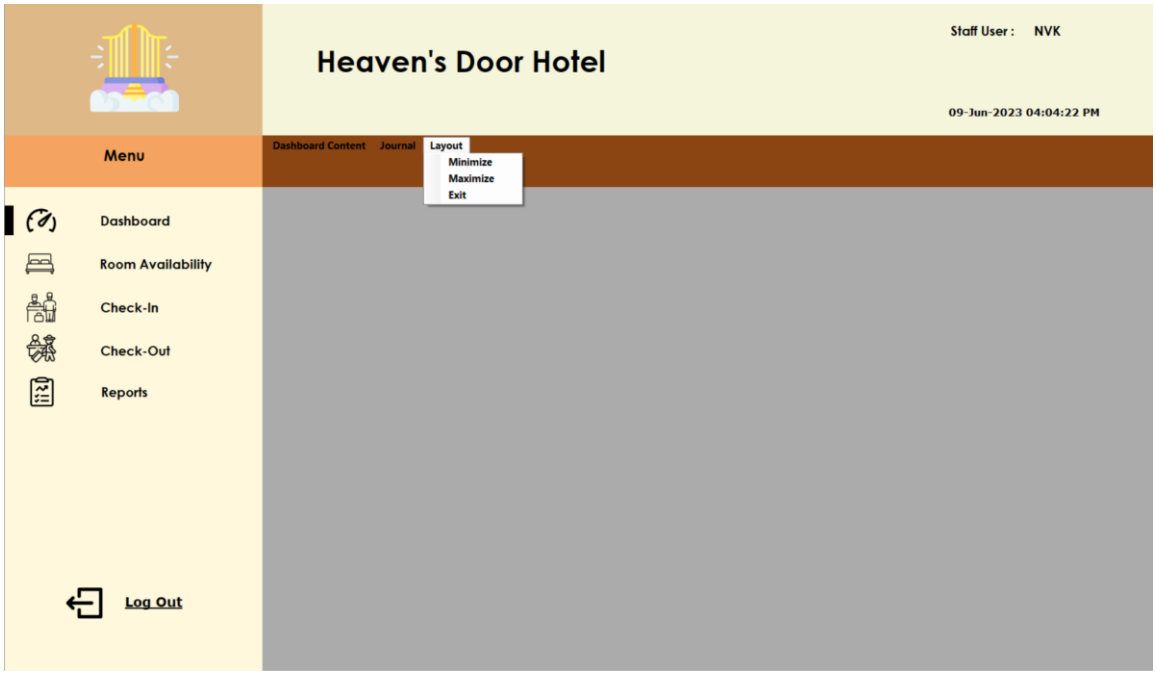


Selain itu ada juga pilihan form *room detail* pada menu *open* dimana setelah menu tersebut di klik akan membuka mdi form yang berguna untuk *read*, *edit*, dan *delete* data yang terdapat di dalam file txt bernama Room Detail.txt. Menu ini berguna bagi resepsionis untuk melihat informasi tentang kamar hotel mencakup tipe, harga, dan juga detail dari kamar tersebut. Resepsionis juga bisa mengubah data tersebut sewaktu-waktu jika terjadi perubahan harga dan lainnya.



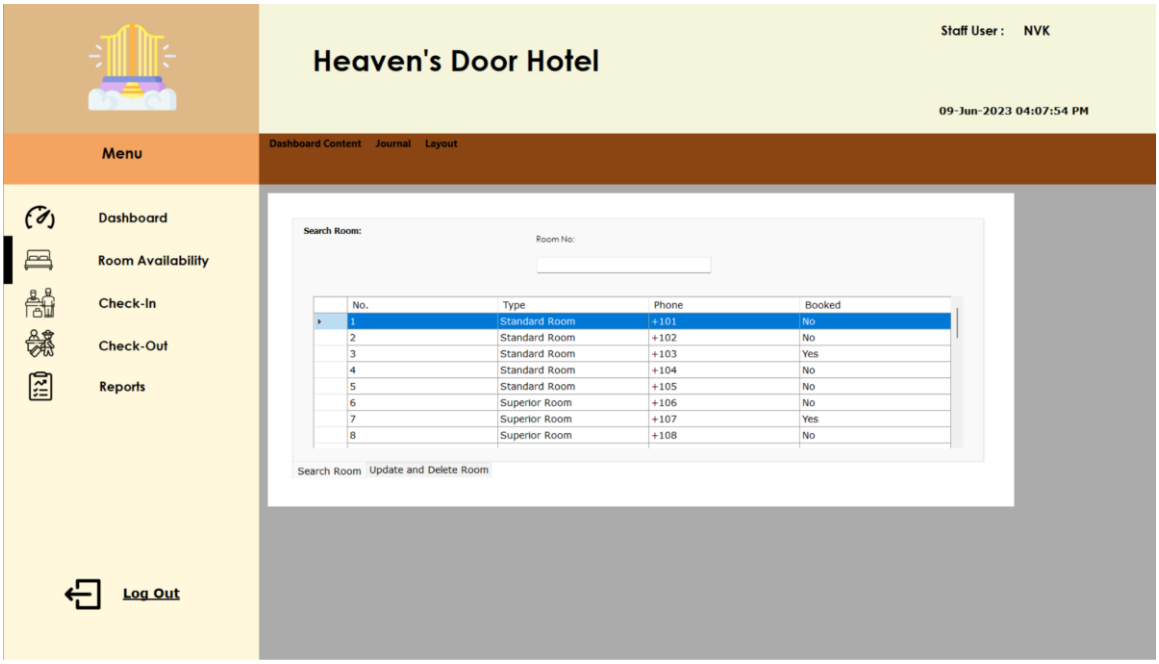
5. Menu Strip - Layout

Menu strip yang terakhir adalah menu strip untuk mengatur tampilan layout dari aplikasi tersebut yaitu *Minimize & Maximize*. Lalu terdapat pilihan exit untuk keluar dari aplikasi secara keseluruhan.



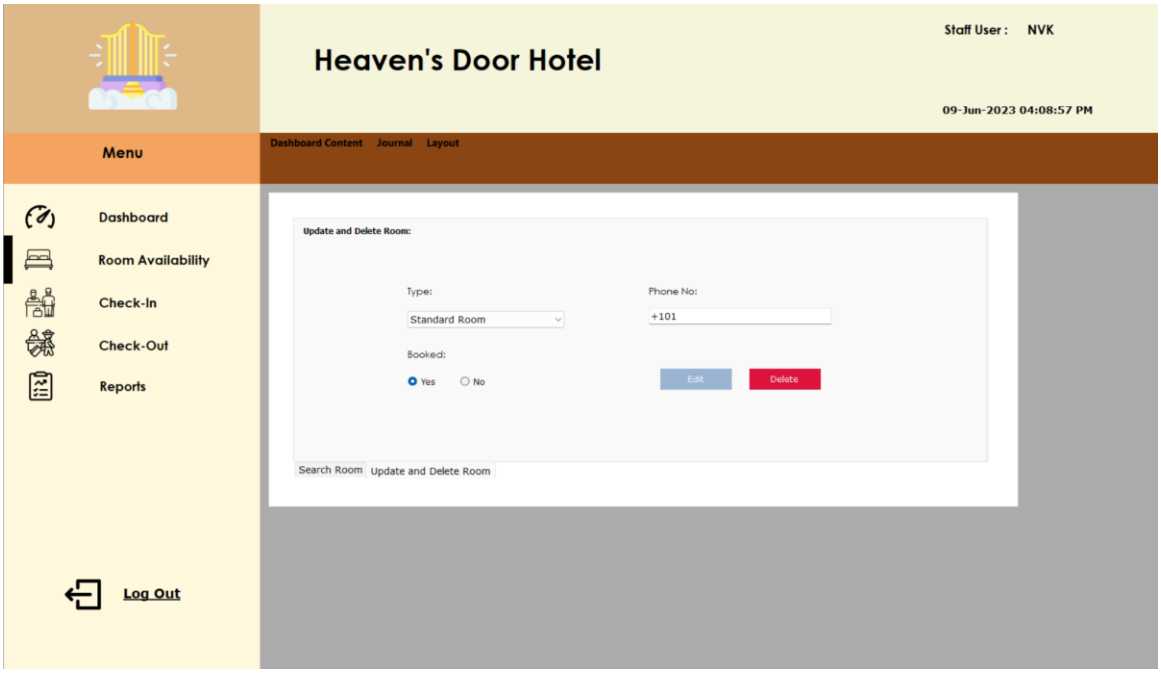
6. Room Availability (Search Room)

Pada page *Room Availability* berguna untuk melihat jumlah ruangan/kamar yang tersedia, melihat data kamar yang ingin dibooking seperti nomor telepon kamar, status *booking* kamar, dan jenis kamar serta nomor kamarnya. Setiap kamar memiliki nomor telepon jika receptionist ingin menelepon tamu yang berada di kamar tersebut.



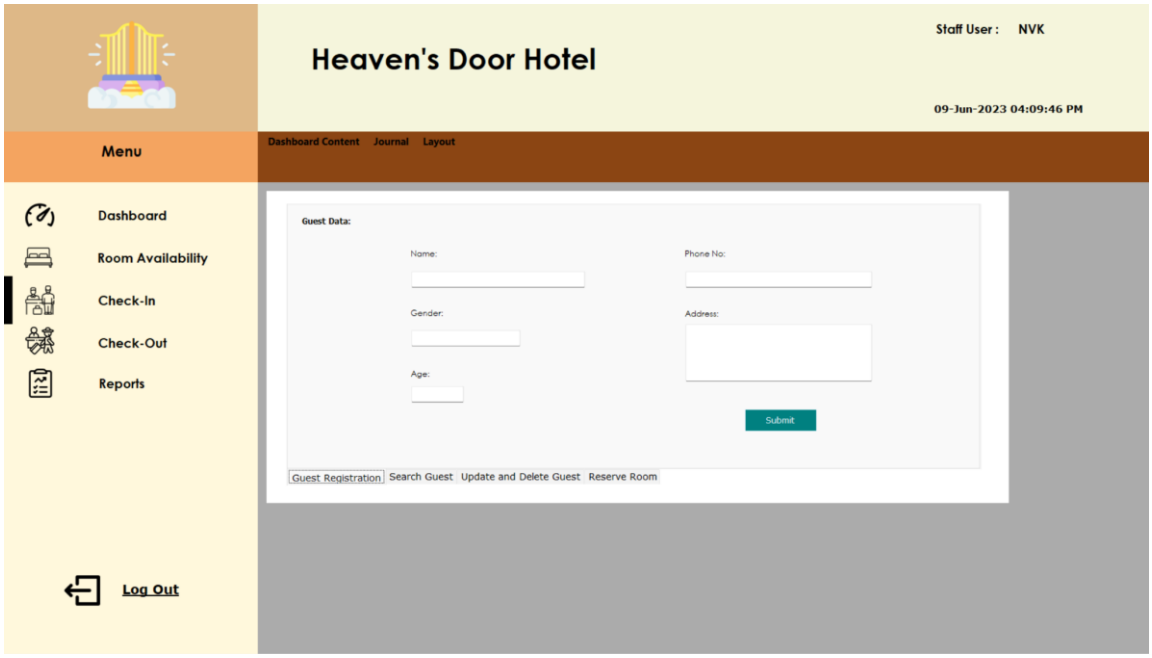
7. Room Availability (Update & Delete Room)

Pada page ini staff bisa melakukan update pada kamar seperti informasi kamar sudah di reservasi atau belum secara manual, nomor telepon dari kamar, dan menghapus kamar dari list *room.* Meskipun bisa melakukannya secara manual, status *booked* pada menu tersebut sudah di program secara otomatis pada menu reservasi dan check-out nantinya.



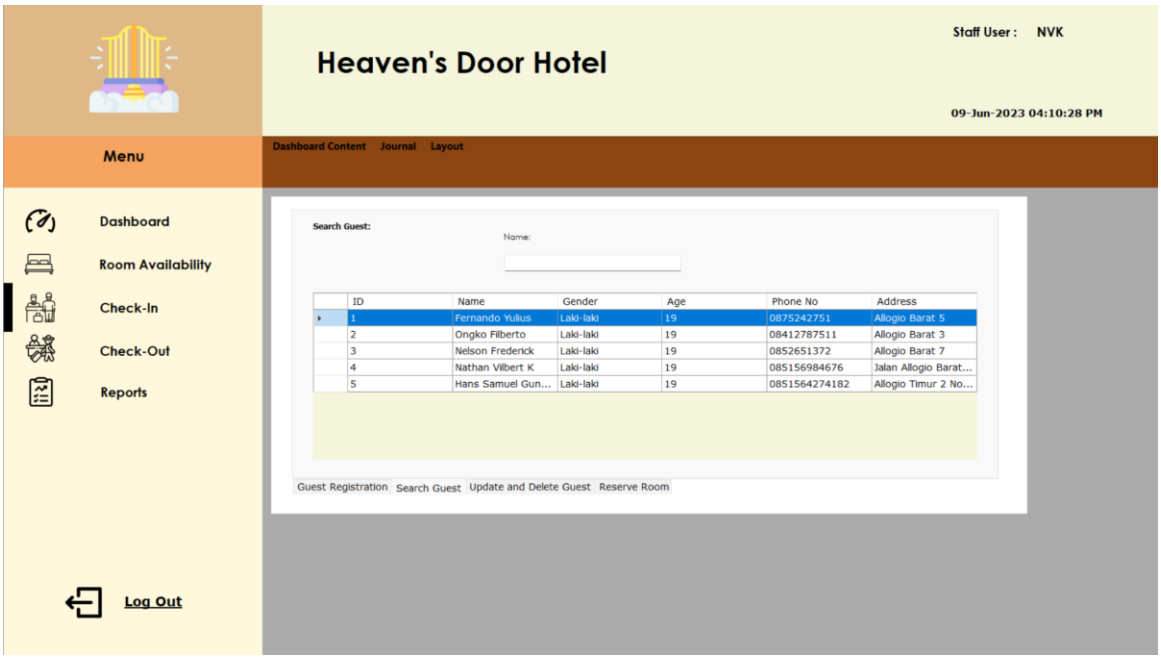
8. Check-In (Guest Registration)

Page ini berguna untuk staff mengisi data dari tamu yang akan melakukan check-in. Sebelum melakukan reservasi, setiap tamu diwajibkan registrasi yang akan dilakukan oleh staff hotel. Guest data akan diisi dengan data tamu seperti nama, jenis kelamin, umur, no HP, dan alamat.



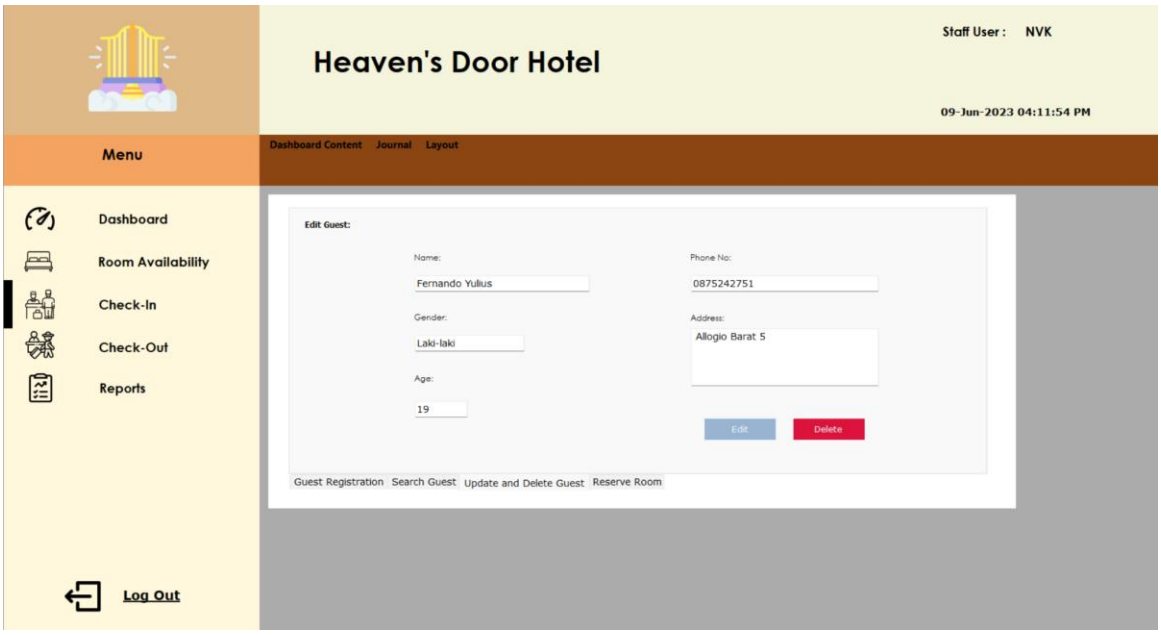
9. Check-In (Search Guest)

Setelah staff hotel me-registrasi data tamu, data-data tamu yang telah terdaftar bisa dilihat melalui tampilan seperti dibawah dan *cell* dari data grid view di bawah bisa dipencet untuk mengambil data secara langsung agar dapat dioper ke pilihan selanjutnya yaitu *update & delete guest* ataupun menu *reserve room.*



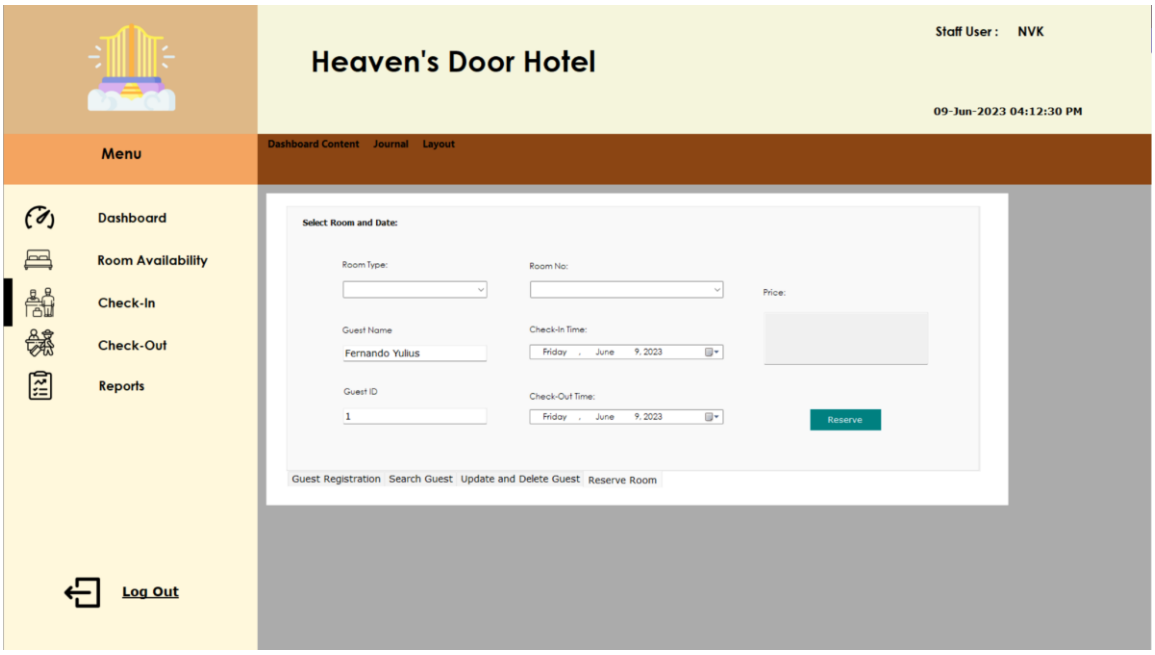
7. Check-In (Update & Delete Guest)

Setelah data guest dialihkan ke sini melalui data grid view, maka staff hotel dapat melakukan perubahan atau menghapus data tamu hotel yang telah melakukan registrasi.



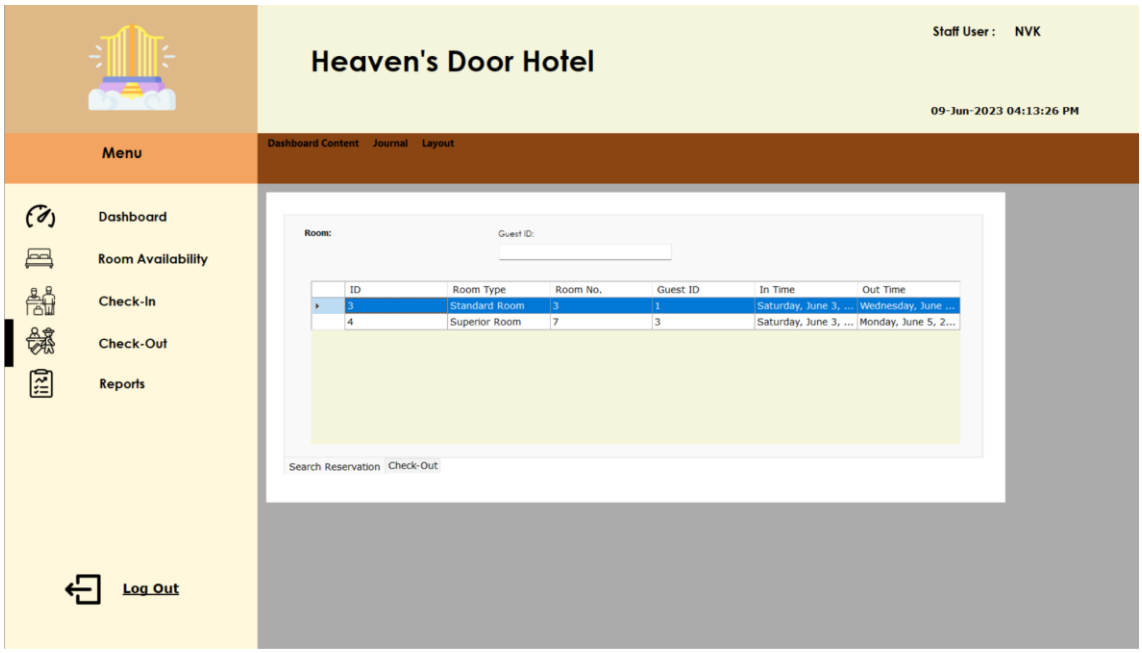
8. Check-In (Reservation)

Menu ini untuk melakukan reservasi kamar dimana data yang akan diminta adalah tipe kamar yang diinginkan oleh tamu, nama tamu, id dari tamu, nomor kamar, serta waktu penginapan yang diajukan oleh tamu hotel. Setelah semua data terisi, kotak harga akan menampilkan total harga yang akan dibayar oleh tamu untuk melakukan reservasi.



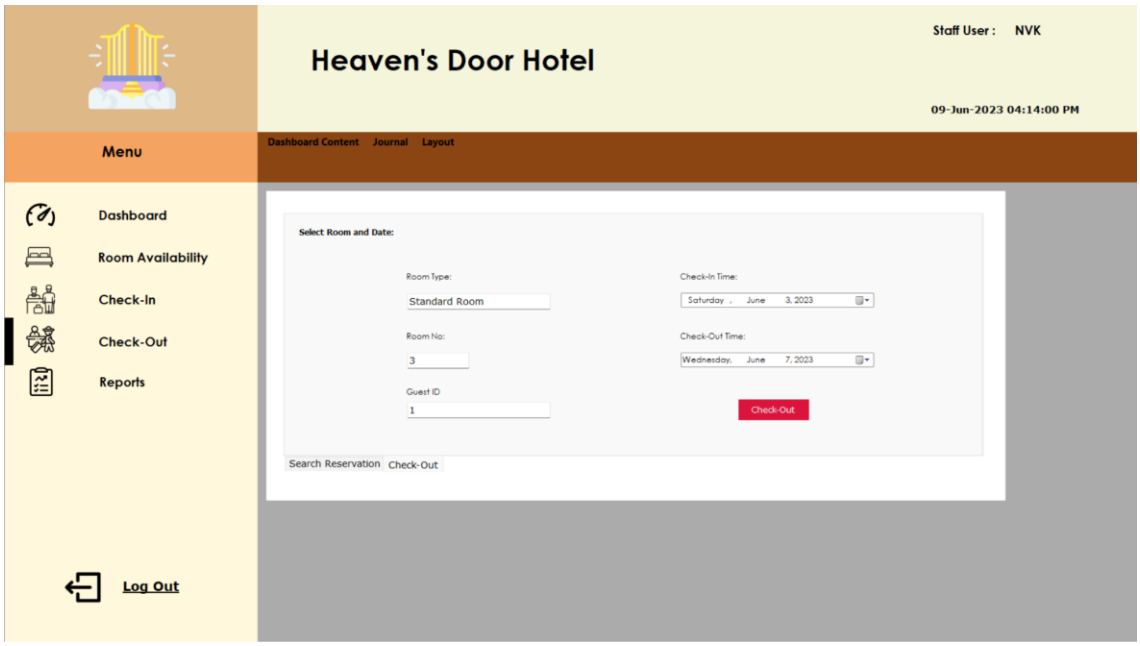
9. Check-Out (Search Reservation)

Jika tamu sudah melakukan reservasi, data-data pada menu reservasi akan dipindah ke dalam database dan akan ditampilkan melalui data grid view seperti pada gambar dibawah. *Cell* dari data grid view dapat di klik untuk mengoper data ke menu selanjutnya yaitu check-out.



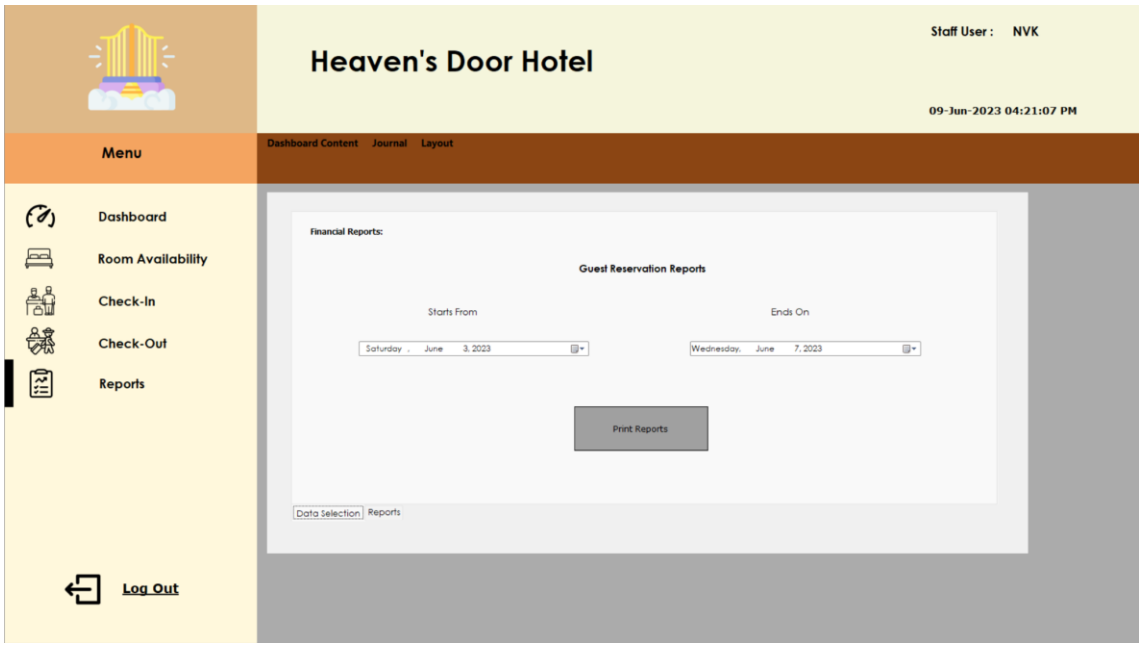
10. Check-Out

Pada page *check-out* berguna untuk ketika penginap akan melakukan check-out. Di page ini staff hotel akan menginput data kamar seperti room type, room number, guest ID, check-in Time, dan check-out Time. Fitur ini berguna untuk memasukkan data history kamar agar kamar memiliki status kosong. Lalu setelah itu akan muncul data harga pada page *reports*.

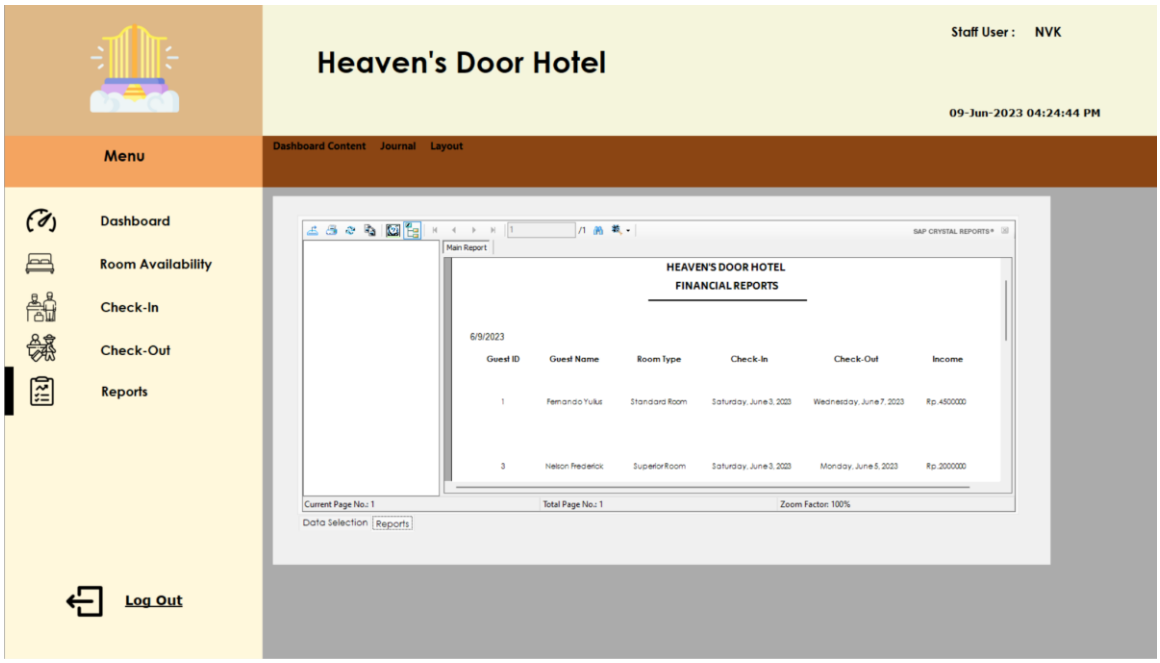


11. Reports Filtering (Crystal Report)

Pada menu report ini, staff hotel akan mendapatkan laporan keuangan hotel seperti data tamu, tipe kamar, nomor kamar, waktu check-in dan check-out serta harga sewa yang menjadi pendapatan bagi hotel. Sebelum mencetak laporannya, akan muncul tampilan untuk memberi resepsionis opsi dalam memilih batas dari waktu check-in dan check-out untuk mencetak laporannya.

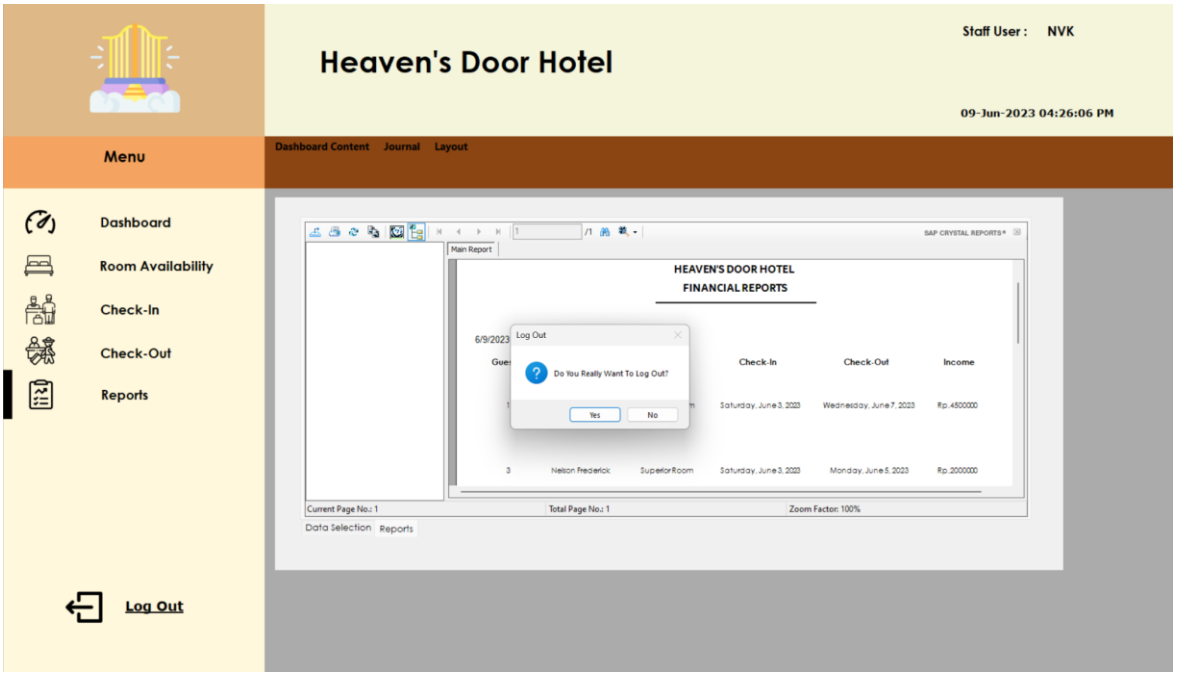


Setelah tanggal sudah ditentukan, resepsionis akan mengklik tombol “Print Reports” lalu layar akan langsung berpindah ke tabpage ke-2 dan langsung mencetak Crystal Report yang sudah di filter berdasarkan pilihan resepsionis.



12. Log Out Staff Account

Pada page ini berguna ketika staff hotel ingin melakukan *log out* setelah melakukan pendataan. Dalam tampilan akan muncul tombol dengan pertanyaan “Do you Really want to Log Out?” dengan opsi tombol *Yes* jika staff hotel ingin benar-benar ingin melakukan log out, tombol *No* jika staff hotel tidak jadi melakukan log out.



2.7 Application Coding

2.7.1. Login Page

```
public Form1()
{
    InitializeComponent();

}
    public string db = "Data Source=DESKTOP-J8PP8MD;Initial Catalog=HeavensDoor;Integrated Security=True";
```

```
private void button1_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection(db);
    con.Open();
    SqlCommand command = new SqlCommand("SELECT * FROM User_Table", con);
    SqlDataReader reader = command.ExecuteReader();

    bool check = false;

    while (reader.Read())
    {
        string username = reader["User_Name"].ToString();
        string password = reader["User_Password"].ToString();

        if (textBoxUsername.Text.Trim() == username && textBoxPassword.Text.Trim() == password)
        {
            check = true;
            break;
        }
    }

    if (textBoxUsername.Text.Trim() == string.Empty || textBoxPassword.Text.Trim() == string.Empty)
    {
        MessageBox.Show("Please Fill The Field.", "Required Field", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        if (check)
        {
            FormDashboard fd = new FormDashboard();
            fd.Username = textBoxUsername.Text;
            textBoxUsername.Clear();
            textBoxPassword.Clear();
            fd.Show();
        }
        else
        {
            MessageBox.Show("Invalid Username/Password.", "Username/Password", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    con.Close();

}
```

kode tersebut digunakan untuk memvalidasi kredensial pengguna/staff selama proses login. Itu terhubung ke database, mengambil informasi pengguna dari tabel yang disebut "User_Table", dan membandingkan nama pengguna dan kata sandi yang dimasukkan dengan nilai dalam tabel. Jika kredensial cocok, itu akan membuka formulir baru (mungkin dasbor) dan menampilkannya. Jika kredensial tidak valid atau jika bidang nama pengguna atau kata sandi kosong, pesan kesalahan yang sesuai akan ditampilkan.

```
private void pictureBoxShow_Click(object sender, EventArgs e)
{
    pictureBoxShow.Hide();
    textBoxPassword.UseSystemPasswordChar = false;
    pictureBoxHidden.Show();
}

1 reference
private void pictureBoxShow_MouseHover(object sender, EventArgs e)
{
    toolTip1.SetToolTip(pictureBoxShow, "Show Password");
}

1 reference
private void pictureBoxHidden_Click(object sender, EventArgs e)
{
    pictureBoxHidden.Hide();
    textBoxPassword.UseSystemPasswordChar = true;
    pictureBoxShow.Show();
}

1 reference
private void pictureBoxHidden_MouseHover(object sender, EventArgs e)
{
        toolTip1.SetToolTip(pictureBoxHidden, "Hide Password");
}
```

Kode tersebut untuk membuat password yang diketik bersifat tertutup (*hidden*) atau terbuka (*show*) dimana terdapat 2 picturebox yang berbeda gambarnya. Ketika picturebox bergambar hidden dipencet maka picturebox itu sendiri akan ter hide dan picturebox lainnya akan di show. Selain itu status dari UseSystemPasswordChar akan diubah menjadi true dan false jika picturebox tersebut di klik.

2.7.2. Dashboard Page

```
1 reference
private void timer1_Tick(object sender, EventArgs e)
{
    labeldate.Text = DateTime.Now.ToString("dd-MMM-yyyy hh:mm:ss tt");
    labelUsername.Text = Username;
}

1 reference
private void button1_Click(object sender, EventArgs e)
{
    MovePanel(button1);
    userControlClient1.Hide();
    userControlRoom1.Hide();
    userControlCheckOut1.Hide();
    userControlFilter1.Hide();
    userControlDashboard1.Show();
}

1 reference
private void button2_Click(object sender, EventArgs e)
{
    MovePanel(button2);
    userControlClient1.Hide();
    userControlRoom1.Show();
    userControlRoom1.Clear();
    userControlRoom1.PerformLayout();
    userControlCheckOut1.Hide();
    userControlFilter1.Hide();
    userControlDashboard1.Hide();
}
```

```
1 reference
private void button3_Click(object sender, EventArgs e)
{
    MovePanel(button3);
    userControlClient1.Clear();
    userControlClient1.Show();
    userControlRoom1.Hide() ;
    userControlCheckOut1.Hide();
    userControlFilter1.Hide();
    userControlDashboard1.Hide();
}

1 reference
private void button4_Click(object sender, EventArgs e)
{
    MovePanel(button4);
    userControlClient1.Hide();
    userControlRoom1.Hide();
    userControlCheckOut1.Show();
    userControlFilter1.Hide();
    userControlDashboard1.Hide();
}

1 reference
private void button5_Click(object sender, EventArgs e)
{
    MovePanel(button5);
    userControlClient1.Hide();
    userControlRoom1.Hide();
    userControlCheckOut1.Hide();
    userControlDashboard1.Hide();
    userControlFilter1.Show();
}
```

Kode diatas berguna untuk menampilkan usercontrol dari tiap menu setiap kali tombol menu diklik. Jika tombol menu pertama diklik maka usercontrol yang akan ditampilkan adalah untuk usercontrol pada menu pertama yaitu dashboard. Ketika 1 usercontrol tersebut ditampilkan maka sisa dari usercontrol yang ditampilkan akan di *hide* dari layar user. Selain itu pada kode diatas terdapat timer tick yang menggunakan toolbox label untuk menampilkan waktu terkini saat user membuka programnya.

```
1 reference
private void showToolStripMenuItem_Click(object sender, EventArgs e)
{
    userControlDashboard1.Show();
}

1 reference
private void minimizeToolStripMenuItem_Click(object sender, EventArgs e)
{
    WindowState = FormWindowState.Minimized;

}

0 references
private void guestNumberToolStripMenuItem_Click(object sender, EventArgs e)...

1 reference
private void hideToolStripMenuItem_Click(object sender, EventArgs e)
{
    userControlDashboard1.Hide();
}

0 references
private void roomAvailableToolStripMenuItem1_Click(object sender, EventArgs e)...

1 reference
private void maximizeToolStripMenuItem_Click(object sender, EventArgs e)
{
    WindowState = FormWindowState.Maximized;
}

0 references
private void exitToolStripMenuItem_Click(object sender, EventArgs e)...

1 reference
private void exitToolStripMenuItem_Click_1(object sender, EventArgs e)
{
    this.Close();
}
```

Kode diatas berguna untuk menampilkan atau tidak menampilkan konten dari usercontroldashboard disaat menu strip dashboard content dipilih. Diatas dan dibawahnya juga merupakan kode untuk menu strip apabila menu strip layout dipilih untuk mengatur size dari layout aplikasi seperti minimized, maximized, dan exit.

```
private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    FormChild formChild = new FormChild();
    formChild.MdiParent = this;



    int parentWidth = this.ClientSize.Width;
    int parentHeight = this.ClientSize.Height;

    int childWidth = formChild.Width;
    int childHeight = formChild.Height;

    int childX = (parentWidth - childWidth) / 2 + 50;
    int childY = (parentHeight - childHeight) / 2 + 50;

    formChild.StartPosition = FormStartPosition.Manual;
    formChild.Location = new Point(childX, childY);

    formChild.Show();



}

1 reference
private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    FormChild2 formChild = new FormChild2();
    formChild.MdiParent = this;

    int parentWidth = this.ClientSize.Width;
    int parentHeight = this.ClientSize.Height;

    int childWidth = formChild.Width;
    int childHeight = formChild.Height;

    int childX = (parentWidth - childWidth) / 2 + 50;
    int childY = (parentHeight - childHeight) / 2 + 50;

    formChild.StartPosition = FormStartPosition.Manual;
    formChild.Location = new Point(childX, childY);

    formChild.Show();
}
```

```
private void guestWishlistToolStripMenuItem_Click(object sender, EventArgs e)
{
    FormChildWishlist formChild = new FormChildWishlist();
    formChild.MdiParent = this;

    int parentWidth = this.ClientSize.Width;
    int parentHeight = this.ClientSize.Height;

    int childWidth = formChild.Width;
    int childHeight = formChild.Height;

    int childX = (parentWidth - childWidth) / 2 + 50;
    int childY = (parentHeight - childHeight) / 2 + 50;

    formChild.StartPosition = FormStartPosition.Manual;
    formChild.Location = new Point(childX, childY);

    formChild.Show();
}

0 references
private void closeToolStripMenuItem_Click(object sender, EventArgs e)
{

}

1 reference
private void roomDetailToolStripMenuItem1_Click(object sender, EventArgs e)
{
    FormChildRoom formChild = new FormChildRoom();
    formChild.MdiParent = this;

    int parentWidth = this.ClientSize.Width;
    int parentHeight = this.ClientSize.Height;

    int childWidth = formChild.Width;
    int childHeight = formChild.Height;

    int childX = (parentWidth - childWidth) / 2 + 50;
    int childY = (parentHeight - childHeight) / 2 + 50;

    formChild.StartPosition = FormStartPosition.Manual;
    formChild.Location = new Point(childX, childY);

    formChild.Show();
}
```

Kedua gambar diatas menunjukan kode pada form dasboard dimana ketika menu strip “Journal” di klik dan di pilih akan menampilkan berbagai mdi form pada layar dashboard seperti *memo pad*, *guest wishlist*, dan *room detail* beserta size spesifik dan penempatannya pada layar dashboard. MDI Form yang akan ditampilkan berasal dari 4 form yang dibuat di luar formdashboard. Agar tampilan dari 4 MDI form tersebut rapih dan memiliki size yang seukuran, maka dibuatlah panjang, lebar, serta posisi dari setiap MDI form yang menggunakan parent form nya sebagai referensi. Lokasinya pun diatur agar berada di tengah-tengah parent formnya.

2.7.3. User Control Dashboard

```
public partial class UserControlDashboard : UserControl
{
    public string Username;


    public UserControlDashboard()
    {
        InitializeComponent();
    }
    public string db = "Data Source = DESKTOP-J8PP8MD; Initial Catalog = HeavensDoor; Integrated Security = True";

    private void UserControl1_Load(object sender, EventArgs e)
    {
        string query = "SELECT COUNT(*) FROM Guest_Table";

        using (SqlConnection connection = new SqlConnection(db))
        {
            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();
            int guestCount = (int)command.ExecuteScalar();
            connection.Close();
            labelUsername.Text = (guestCount.ToString());
        }
        string query2 = "SELECT COUNT(*) FROM Room_Table WHERE Room_Booked = 'No'";

        using (SqlConnection connection = new SqlConnection(db))
        {
            SqlCommand command = new SqlCommand(query2, connection);
            connection.Open();
            int roomCount = (int)command.ExecuteScalar();
            connection.Close();
            labelroom.Text = (roomCount.ToString());
        }

        }
```

Kode diatas untuk menunjukan angka banyaknya tamu yang terdapat di dalam hotel dengan mengambil data pada Guest_Table dengan menghitung jumlah baris data yang ada dan valuenya akan di oper ke labelUsername dalam bentuk string. Dibawahnya juga kode untuk menunjukan seberapa banyak kamar yang masih kosong pada hotel dengan membaca isi dari kolom tabel Room_Tabel yaitu kolom Room_Booked dan menghitung banyaknya baris dari Room_Booked yang memiliki value “No”. Kemudian kedua angka tersebut ditunjukkan melalui labelUsername dan labelroom.

2.7.4. User Control Room

```
namespace HotelReceptionistsSystem.User_Control
{
    4 references
    public partial class UserControlRoom : UserControl
    {
        1 reference
        public UserControlRoom()
        {
            InitializeComponent();
        }
        private string No = "", Free = "";

        public string db = "Data Source = DESKTOP-J8PP8MD; Initial Catalog = HeavensDoor; Integrated Security = True";
        1 reference
        private void UserControlRoom_Load(object sender, EventArgs e)
        {
            string query = "SELECT * FROM Room_Table";

            try
            {
                using (SqlConnection connection = new SqlConnection(db))
                {
                    connection.Open();

                    SqlCommand command = new SqlCommand(query, connection);
                    SqlDataAdapter adapter = new SqlDataAdapter(command);
                    DataTable dataTable = new DataTable();
                    adapter.Fill(dataTable);

                    dataGridViewRoom.DataSource = dataTable;
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Terjadi kesalahan saat menampilkan data: " + ex.Message);
            }
        }
```

Kode diatas bertujuan untuk menampilkan seluruh isi dari tabel Room_Table yang berasal dari database HeavensDoor dan menampilkannya melalui datagridviewRoom. Ketika pembacaan gagal maka akan muncul message box yang berisi "Terjadi kesalahan saat menampilkan data". Seluruh data yang ditampilkan akan mengikuti format pada datagridview dimana property value dari kolomnya telah disesuaikan dengan nama dari kolom pada tabel Room_Table di dalam database HeavensDoor.

```
private void tabPageSearchRoom_Enter(object sender, EventArgs e)
{
    string query = "SELECT * FROM Room_Table";

    try
    {
        using (SqlConnection connection = new SqlConnection(db))
        {
            connection.Open();

            SqlCommand command = new SqlCommand(query, connection);
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            dataGridViewRoom.DataSource = dataTable;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Terjadi kesalahan saat menampilkan data: " + ex.Message);
    }

}

1 reference
private void textBoxSearchName_TextChanged(object sender, EventArgs e)
{
    string query = "SELECT * FROM Room_Table WHERE Room_Phone LIKE '%" + textBoxSearchNo.Text + "%'";

    try
    {
        using (SqlConnection connection = new SqlConnection(db))
        {
            connection.Open();

            SqlCommand command = new SqlCommand(query, connection);
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            dataGridViewRoom.DataSource = dataTable;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Terjadi kesalahan saat menampilkan data: " + ex.Message);
    }

}
```

Kode diatas untuk menunjukan kembali seluruh isi dari tabel Room_Table pada tabpage yang berbeda. Setelah menampilkan seluruh data yang terdapat di dalam Room_Table ke dalam datagridview, kode dibawahnya bertujuan untuk mencarikan data yang sesuai dengan input dari staff hotel nantinya dimana kunci dari pencarian tersebut didasari oleh value dari textboxSearchNo yang merepresentasikan nomor telepon dari tiap kamar. Ketika value textbox sama dengan value pada kolom database, maka sebaris dari data yang dicari akan dimunculkan melalui dataGridViewRoom.

```
1 reference
private void buttonDelete_Click(object sender, EventArgs e)
{
    if (comboBoxType.SelectedIndex == -1 || textBoxPhoneNo.Text.Trim() == string.Empty || Free == "")

    {
        MessageBox.Show("Please Fill The Field.", "Required Field", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    else
    {
        string query = "DELETE FROM Room_Table WHERE Room_Phone = @Phone";

        try
        {
            using (SqlConnection connection = new SqlConnection(db))
            {
                connection.Open();

                SqlCommand command = new SqlCommand(query, connection);
                command.Parameters.AddWithValue("@Phone", textBoxPhoneNo.Text.Trim());

                int rowsAffected = command.ExecuteNonQuery();

                if (rowsAffected > 0)
                {
                    MessageBox.Show("Delete Room Succesfuly Done!");
                }
                else
                {
                    MessageBox.Show("Failed To Delete. Please Try Again.");
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error Happens: " + ex.Message);
        }
    }
}
```

Kode diatas berfungsi sebagai perintah delete. Delete pada perintah kode diatas akan menghapus data dari tabel Room_Table sesuai dengan room yang dipilih oleh user dengan memanfaatkan nomor telepon dari kamar yang dipilih sebagai kunci dalam mencari data di database. Ketika value dari textboxPhoneNo sama dengan value dari kolom Room_Phone maka sebaris data yang ingin di *delete* akan terhapus dari database HeavensDoor.

```
1 reference
private void buttonEdit_Click(object sender, EventArgs e)
{
    if (radioButtonNo.Checked = true)
    {
        Free = "Yes";
    }
    if (radioButtonYes.Checked = true)
    {
        Free = "No";
    }

    if (comboBoxType.SelectedIndex == -1 || textBoxPhoneNo.Text.Trim() == string.Empty || Free == "")

    {
        MessageBox.Show("Please Fill The Field.", "Required Field", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        string query = "UPDATE Room_Table SET Room_Type = @Type, Room_Phone = @Phone, Room_Booked = @Booked WHERE Room_Phone = @Phone";

        try
        {
            using (SqlConnection connection = new SqlConnection(db))
            {
                connection.Open();

                SqlCommand command = new SqlCommand(query, connection);
                command.Parameters.AddWithValue("@Type", comboBoxType.SelectedItem.ToString().Trim());
                command.Parameters.AddWithValue("@Phone", textBoxPhoneNo.Text.Trim());
                command.Parameters.AddWithValue("@Booked", Free);


                int rowsAffected = command.ExecuteNonQuery();

                if (rowsAffected > 0)
                {
                    MessageBox.Show("Room Succesfully Updated!");
                }
                else
                {
                    MessageBox.Show("Failed To Update Room. Please Try Again.");
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error Happens: " + ex.Message);
        }
```

Kode diatas adalah kode untuk mengubah value dari string "Free" dimana ketika radiobutton dipencet No maka status available dari kamar tersebut adalah Yes, begitu juga sebaliknya. Serangkaian kode dibawahnya untuk melakukan edit atau update pada room dimana data yang bisa di edit sesuai urutan adalah tipe dari kamar, nomor telepon kamar, dan status *booking* dari kamar tersebut. Kunci dalam mencari data tersebut didasarkan oleh kolom Room_Phone dimana valuenya akan dicari melalui textboxPhoneNo oleh user.

```
private void dataGridViewRoom_CellClick_1(object sender, DataGridViewCellEventArgs e)
{
        if(e.RowIndex != -1)
    {
        DataGridViewRow row = dataGridViewRoom.Rows[e.RowIndex];
        No = row.Cells[0].Value.ToString();
        comboBoxType.SelectedItem = row.Cells[1].Value.ToString();
        textBoxPhoneNo.Text = row.Cells[2].Value.ToString();
        Free = row.Cells[3].Value.ToString();
        if (Free == "Yes")
            radioButtonNo.Checked = true;
        if (Free == "No")
            radioButtonYes.Checked = true;

    }
}
```

Kode diatas dibuat agar setiap kali user berinteraksi dengan datagridview dengan mengklik sel dari tabel datagridview, maka sebaris data yang dipilih dari tabel tersebut akan dioper ke masing-masing combobox, textbox, serta value string "Free" yang akan dipakai di tabpage selanjutnya karena toolbox-toolbox tersebut berada di tab page berikutnya.

2.7.5. User Control Client

```
0 references
public void DisplayAndSearch(string query, DataGridView dataGridView)
{
    try
    {
        using (SqlConnection con = new SqlConnection(db))
        {
            con.Open();

            SqlCommand command = new SqlCommand(query, con);
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            dataGridView.DataSource = dataTable;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Terjadi kesalahan saat menampilkan data: " + ex.Message);
    }
}
```

Ini adalah kode untuk mengambil data dari database dan menampilkannya di DataGridView. Kode tersebut terdiri dari beberapa langkah. Pertama, koneksi ke database dibuka menggunakan objek SqlConnection. Kemudian, sebuah objek SqlCommand dibuat dengan query yang diberikan. Hasil eksekusi query diambil dan disimpan dalam objek DataTable menggunakan SqlDataAdapter. Setelah itu, objek DataTable digunakan sebagai sumber data untuk DataGridView. Dengan mengatur properti DataSource DataGridView ke DataTable, data dari database akan ditampilkan dalam DataGridView. Jika terjadi kesalahan saat menjalankan kode, misalnya koneksi ke database gagal atau query tidak valid, pesan kesalahan akan ditampilkan melalui MessageBox. Kode tersebut memberikan cara yang sederhana dan efektif untuk menampilkan data dari database ke dalam tampilan DataGridView.

```
private void buttonSubmit_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection(db);
    con.Open();

    string insertQuery = "INSERT INTO Guest_table (Guest_Name, Guest_Gender, Guest_Age, Guest_Phone, Guest_Address) VALUES (@Name, @Gender, @Age, @Phone, @Address)";
    SqlCommand insertCommand = new SqlCommand(insertQuery, con);
    insertCommand.Parameters.AddWithValue("@Name", textBoxName.Text.Trim());
    insertCommand.Parameters.AddWithValue("@Gender", textBoxGender.Text.Trim());
    insertCommand.Parameters.AddWithValue("@Age", textBoxAge.Text.Trim());
    insertCommand.Parameters.AddWithValue("@Phone", textBoxPhone.Text.Trim());
    insertCommand.Parameters.AddWithValue("@Address", textBoxAddress.Text.Trim());

    int rowsAffected = insertCommand.ExecuteNonQuery();


    if (textBoxName.Text.Trim() == string.Empty || textBoxGender.Text.Trim() == string.Empty || textBoxAge.Text.Trim() == string.Empty
        || textBoxPhone.Text.Trim() == string.Empty || textBoxAddress.Text.Trim() == string.Empty)
    {
        MessageBox.Show("Please Fill The Field.", "Required Field", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {

        if (rowsAffected > 0)
        {
            MessageBox.Show("Guest Registration Succesfully Done!");
        }
        else
        {
            MessageBox.Show("Guest Registration Failed. Please Try Again.");
        }


    }
```

Kode diatas yang berada pada public void DisplayAndSearch adalah kode untuk menampilkan data dari dataTable ke dalam dataGridView. Lalu kode yang berada pada public void buttonSubmit_Click adalah kode yang berfungsi sebagai perintah insert yang berguna untuk memasukkan value ke dalam tabel Guest_Table sesuai urutan valuenya yang diambil dari tiap textbox pada usercontrolClient.

```
private void tabPage2_Enter(object sender, EventArgs e)
{
    string query = "SELECT * FROM Guest_Table";

    try
    {
        using (SqlConnection connection = new SqlConnection(db))
        {
            connection.Open();

            SqlCommand command = new SqlCommand(query, connection);
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            dataGridViewGuest.DataSource = dataTable;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Terjadi kesalahan saat menampilkan data: " + ex.Message);
    }

}

1 reference
private void tabPage2_Leave(object sender, EventArgs e)
{
    textBoxSearchName.Clear();
}
```

Kode diatas adalah perintah untuk menampilkan seluruh data dari tabel Guest_Table dan menampilkannya pada dataGridView. Kode tersebut akan jalan ketika tabpage ke 2 dari

UserControlClient di klik atau dimasuki oleh user dan datagridview akan secara otomatis terpenuhi oleh data tabel Guest_Table sesuai urutan dari kolomnya. Dan ketika tabpage ke 2 ditinggalkan oleh user maka value dari textboxSearchName akan dibersihkan.

```
private void textBoxSearchName_TextChanged(object sender, EventArgs e)
{
    string query = "SELECT * FROM Guest_Table WHERE Guest_Name LIKE '%" + textBoxSearchName.Text + "%'";

    try
    {
        using (SqlConnection connection = new SqlConnection(db))
        {
            connection.Open();

            SqlCommand command = new SqlCommand(query, connection);
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            dataGridViewGuest.DataSource = dataTable;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Terjadi kesalahan saat menampilkan data: " + ex.Message);
    }

}
```

Kode tersebut berfungsi sebagai filter setiap kali teks dari textboxsearchname terganti agar data dari tabel Guest_Table yang tertampil sesuai dengan value dari textboxSearchName pada usercontrolClient. Ketika valuenya sama dengan kolom Guest_Name maka data yang dicari akan dimunculkan.

```
private void buttonEdit_Click(object sender, EventArgs e)
{


    if (textBoxName1.Text.Trim() == string.Empty || textBoxGender1.Text.Trim() == string.Empty || textBoxAge1.Text.Trim() == string.Empty
      || textBoxPhone1.Text.Trim() == string.Empty || textBoxAddress1.Text.Trim() == string.Empty)
    {
        MessageBox.Show("Please Fill The Field.", "Required Field", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {


        string query = "UPDATE Guest_Table SET Guest_Name = @Name, Guest_Gender = @Gender, Guest_Age = @Age, Guest_Phone = @Phone," +
            " Guest_Address = @Address WHERE Guest_Name = @Name";

        try
        {
            using (SqlConnection connection = new SqlConnection(db))
            {
                connection.Open();

                SqlCommand command = new SqlCommand(query, connection);
                command.Parameters.AddWithValue("@Name", textBoxName1.Text.Trim());
                command.Parameters.AddWithValue("@Gender", textBoxGender1.Text.Trim());
                command.Parameters.AddWithValue("@Age", textBoxAge1.Text.Trim());
                command.Parameters.AddWithValue("@Phone", textBoxPhone1.Text.Trim());
                command.Parameters.AddWithValue("@Address", textBoxAddress1.Text.Trim());


                int rowsAffected = command.ExecuteNonQuery();

                if (rowsAffected > 0)
                {
                    MessageBox.Show("Data Succesfully Updated!");
                }
                else
                {
                    MessageBox.Show("Failed To Update Data. Please Try Again.");
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error Happens: " + ex.Message);
        }


    }
```

Kode tersebut berfungsi sebagai perintah update pada Guest_Table sesuai dengan setiap urutan valuenya dan parameter dari tiap textbox yang diisi pada kode diatas. Data yang di update akan langsung terganti pada database HeavensDoor dan pesan berhasil akan muncul.

```
1 reference
private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    if (e.RowIndex != -1)
    {
        DataGridViewRow row = dataGridViewGuest.Rows[e.RowIndex];

        textBoxName1.Text = row.Cells[1].Value.ToString();
        textBoxGender1.Text = row.Cells[2].Value.ToString();
        textBoxAge1.Text = row.Cells[3].Value.ToString();
        textBoxPhone1.Text = row.Cells[4].Value.ToString();
        textBoxAddress1.Text = row.Cells[5].Value.ToString();
        textBoxGuestID.Text = row.Cells[0].Value.ToString();
        textBoxName2.Text = row.Cells[1].Value.ToString();

    }
}
```

Kode diatas berguna untuk menginput setiap isi atau value dari sel tabel datagridview yang di klik oleh user ke dalam berbagai textbox yang sudah di pastikan. Ketika *cell*

datagridview di klik maka secara otomatis value dari sebaris data tersebut akan di *pass* ke dalam textbox yang ada pada gambar diatas.

```
private void buttonDelete_Click(object sender, EventArgs e)
{
    if (textBoxName1.Text.Trim() == string.Empty || textBoxGender1.Text.Trim() == string.Empty || textBoxAge1.Text.Trim() == string.Empty
        || textBoxPhone1.Text.Trim() == string.Empty || textBoxAddress1.Text.Trim() == string.Empty)
    {
        MessageBox.Show("Please Fill The Field.", "Required Field", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        string query = "DELETE FROM Guest_Table WHERE Guest_Name = @Name";

        try
        {
            using (SqlConnection connection = new SqlConnection(db))
            {
                connection.Open();

                SqlCommand command = new SqlCommand(query, connection);
                command.Parameters.AddWithValue("@Name", textBoxName1.Text.Trim());

                int rowsAffected = command.ExecuteNonQuery();

                if (rowsAffected > 0)
                {
                    MessageBox.Show("Guest Data Deleted!");
                }
                else
                {
                    MessageBox.Show("Failed To Delete Guest Data. Please Try Again.");
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error Happens: " + ex.Message);
        }
    }

}
```

Kode diatas berfungsi sebagai perintah delete pada tabpage update & delete usercontrolClient dimana data yang akan di hapus dari tabel Guest_Table adalah data dengan kolom Guest_Name yang sama dengan textboxName1.

```
private void buttonReserve_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection(db);
    con.Open();

    string insertQuery = "INSERT INTO Reservation_Table (Reservation_Room_Type, Reservation_Room_Number, Reservation_Guest_ID," +
        " Reservation_In, Reservation_Out) VALUES (@Type, @Number, @ID, @In, @Out)";
    SqlCommand insertCommand = new SqlCommand(insertQuery, con);
    insertCommand.Parameters.AddWithValue("@Type", comboBoxType.SelectedItem.ToString().Trim());
    insertCommand.Parameters.AddWithValue("@Number", comboBoxNo.SelectedItem.ToString().Trim());
    insertCommand.Parameters.AddWithValue("@ID", textBoxGuestID.Text.Trim());
    insertCommand.Parameters.AddWithValue("@In", dateTimePickerIn.Text.Trim());
    insertCommand.Parameters.AddWithValue("@Out", dateTimePickerOut.Text.Trim());


    int rowsAffected = insertCommand.ExecuteNonQuery();

    if (comboBoxType.SelectedIndex == -1 || comboBoxNo.SelectedIndex == -1 || textBoxGuestID.Text.Trim() == string.Empty)
    {
        MessageBox.Show("Please Fill The Field.", "Required Field", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        if (rowsAffected > 0)
        {
            string updateQuery = "UPDATE Room_Table SET Room_Booked = 'Yes' WHERE Room_Number = @Number";
            SqlCommand updateCommand = new SqlCommand(updateQuery, con);
            updateCommand.Parameters.AddWithValue("@Number", comboBoxNo.SelectedItem.ToString().Trim());
            updateCommand.ExecuteNonQuery();

            string insertQuery2 = "INSERT INTO Report_Table (Report_Room_Type, Report_Guest_Name, Report_Guest_ID, Report_Room_Number, Report_Check_In, " +
                "Report_Check_Out, Report_Price) VALUES (@Type, @Name, @ID, @Number, @In, @Out, @Price)";
            SqlCommand insertCommand2 = new SqlCommand(insertQuery2, con);
            insertCommand2.Parameters.AddWithValue("@Type", comboBoxType.SelectedItem.ToString().Trim());
            insertCommand2.Parameters.AddWithValue("@Number", comboBoxNo.SelectedItem.ToString().Trim());
            insertCommand2.Parameters.AddWithValue("@Name", textBoxName2.Text.Trim());
            insertCommand2.Parameters.AddWithValue("@ID", textBoxGuestID.Text.Trim());
            insertCommand2.Parameters.AddWithValue("@In", dateTimePickerIn.Text.Trim());
            insertCommand2.Parameters.AddWithValue("@Out", dateTimePickerOut.Text.Trim());
            insertCommand2.Parameters.AddWithValue("@Price", textBoxPrice.Text.Trim());
            insertCommand2.ExecuteNonQuery();

            MessageBox.Show("Reservation Succesfully Done!");
        }
        else
        {
            MessageBox.Show("Reservation Failed. Please Try Again.");
        }
    }

    con.Close();
```

Kode diatas berfungsi sebagai reservation yang memuat perintah insert dan update pada tabel Reservation_Table, Room_Table, dan Report_Table dimana seluruh data yang ada pada tabpage tersebut akan dimasukkan ke dalam tabel Reservation_Table dan Report_Table. Ketika button Reserve di klik maka status dari kamar dengan nomor telepon yang sama dengan table Room_Table, kolom Room_Booked nya akan secara otomatis terganti valuenya menjadi “Yes”.

```
private void dateTimePickerOut_ValueChanged(object sender, EventArgs e)
{
    DateTime selectedDate1 = dateTimePickerOut.Value;
    DateTime selectedDate2 = dateTimePickerIn.Value;
    TimeSpan difference = selectedDate1 - selectedDate2;
    int days = difference.Days;

    if (comboBoxType.SelectedItem == "Standard Room")
    {
        int totalprice = (days * 1500000);
        textBoxPrice.Text = "Rp." + totalprice.ToString();
    }
    else if (comboBoxType.SelectedItem == "Superior Room")
    {
        int totalprice = (days * 2000000);
        textBoxPrice.Text = "Rp." + totalprice.ToString();
    }
    else if (comboBoxType.SelectedItem == "Deluxe Room")
    {
        int totalprice = (days * 2500000);
        textBoxPrice.Text = "Rp." + totalprice.ToString();
    }
    else if (comboBoxType.SelectedItem == "Suite Room")
    {
        int totalprice = (days * 4000000);
        textBoxPrice.Text = "Rp." + totalprice.ToString();
    }
    else if (comboBoxType.SelectedItem == "Twin Room")
    {
        int totalprice = (days * 1700000);
        textBoxPrice.Text = "Rp." + totalprice.ToString();
    }
    else if (comboBoxType.SelectedItem == "Double Room")
    {
        int totalprice = (days * 1500000);
        textBoxPrice.Text = "Rp." + totalprice.ToString();
    }
    else if (comboBoxType.SelectedItem == "Family Room")
    {
        int totalprice = (days * 2200000);
        textBoxPrice.Text = "Rp." + totalprice.ToString();
    }
    else
    {
        MessageBox.Show("Error Calculating Price. Please Try Again.");
    }

    textBoxPrice.BackColor = Color.White;
}
```

Kode diatas berfungsi sebagai penghitung harga sewa dari reservasi dimana kode tersebut mengambil value dari banyaknya hari yang diambil oleh tamu dan tipe kamar yang dipilih oleh tamu. Ketika dua value tersebut sudah ada, maka banyaknya hari yang diinginkan oleh tamu akan dikali dengan harga tipe kamar tersebut. Setelah itu total harga nya akan dimunculkan pada textBoxPrice.

2.7.6. User Control Check-Out

```
private void UserControlCheckOut_Load(object sender, EventArgs e)
{
    string query = "SELECT * FROM Reservation_Table WHERE Reservation_Guest_ID LIKE '%" + textBoxGuestID.Text + "%'";

    try
    {
        using (SqlConnection connection = new SqlConnection(db))
        {
            connection.Open();

            SqlCommand command = new SqlCommand(query, connection);
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            dataGridViewReservation.DataSource = dataTable;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Terjadi kesalahan saat menampilkan data: " + ex.Message);
    }
}
```

Kode diatas bertujuan untuk menampilkan seluruh data yang ada pada tabel Reservation_Table ke dataGridViewReservation yang berada pada UserControlCheckOut dimana fungsinya akan menjadi menu bagi resepsionis untuk mengecek data tamu yang pada waktu tersebut sedang melakukan penginapan.

```
private void textBoxGuestID_TextChanged(object sender, EventArgs e)
{
    string query = "SELECT * FROM Reservation_Table WHERE Reservation_Guest_ID LIKE '%"
        + textBoxGuestID.Text + "%'";

    try
    {
        using (SqlConnection connection = new SqlConnection(db))
        {
            connection.Open();

            SqlCommand command = new SqlCommand(query, connection);
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            dataGridViewReservation.DataSource = dataTable;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Terjadi kesalahan saat menampilkan data: " + ex.Message);
    }
}
```

Kode diatas berguna sebagai pencarian seorang tamu yang sedang melakukan penginapan dengan ID dari tamu tersebut sebagai kunci dalam mencari data dari Reservation_Table. Ketika text dari textboxGuestID terganti valuenya maka dia akan otomatis mencari data yang sesuai dengan ID tamu yang diketik dan secara langsung menampilkan data tersebut pada datagridviewreservation.

```
1 reference
private void dataGridViewReservation_CellClick(object sender, DataGridViewCellEventArgs e)
{
    if(e.RowIndex != -1)
    {
        DataGridViewRow row = dataGridViewReservation.Rows[e.RowIndex];
        RID = row.Cells[0].Value.ToString();
        textBoxRoomType.Text = row.Cells[1].Value.ToString();
        textBoxRoomNumber.Text = row.Cells[2].Value.ToString();
        textBoxGuestID1.Text = row.Cells[3].Value.ToString();
        dateTimePickerIn.Text = row.Cells[4].Value.ToString();
        dateTimePickerOut.Text = row.Cells[5].Value.ToString();
    }
}
```

Kode tersebut berguna agar saat sel dari tabel datagridview di klik maka sebaris dari sel tersebut akan dioper datanya ke textbox dan datetimepicker pada tabpage selanjutnya sehingga staff hotel tidak perlu mengetik untuk kedua kalinya dalam memasukkan data tamu.

```
1 reference
private void buttonCheckOut_Click(object sender, EventArgs e)
{
    if (textBoxRoomType.Text.Trim() == string.Empty || textBoxGuestID1.Text.Trim() == string.Empty)

    {
        MessageBox.Show("Please Fill The Field.", "Required Field", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        string query = "DELETE FROM Reservation_Table WHERE Reservation_Guest_ID = @ID";

        try
        {
            using (SqlConnection connection = new SqlConnection(db))
            {
                connection.Open();

                SqlCommand command = new SqlCommand(query, connection);
                command.Parameters.AddWithValue("@ID", textBoxGuestID1.Text.Trim());

                int rowsAffected = command.ExecuteNonQuery();

                if (rowsAffected > 0)
                {
                    string updateQuery = "UPDATE Room_Table SET Room_Booked = 'No' WHERE Room_Number = @Number";
                    SqlCommand updateCommand = new SqlCommand(updateQuery, connection);
                    updateCommand.Parameters.AddWithValue("@Number", textBoxRoomNumber.Text.ToString().Trim());
                    updateCommand.ExecuteNonQuery();
                    MessageBox.Show("Check-Out Succesfully Done!");
                }
                else
                {
                    MessageBox.Show("Failed To Check-Out. Please Try Again.");
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error Happens: " + ex.Message);
        }
    }

}
```

Kode diatas akan berjalan ketika buttoncheckout di klik yang berfungsi sebagai perintah delete data dari tabel database dimana data yang akan di delete adalah data yang memiliki Reservation_Guest_ID yang sama dengan yang diketik atau dipilih oleh staff hotel. Selain

men-delete data tersebut dari tabel Reservation_Table, query akan melakukan update pada Room_Table yang di checkout dimana status dari Room_Booked nya akan diubah valuenya menjadi “No”.

2.7.7. User Control Filter Reports

```
1 reference
private void buttonCheckOut_Click_1(object sender, EventArgs e)
{
    DateTime startDate = dateTimePickerStart.Value;
    DateTime endDate = dateTimePickerEnd.Value;
    string startDateFormat = startDate.ToString("dddd, MMMM d, yyyy");
    string endDateFormat = endDate.ToString("dddd, MMMM d, yyyy");


    ReportDocument reportDocument = new ReportDocument();


    reportDocument.Load("C:\\Users\\User\\Documents\\Visual Studio 2022\\HotelReceptionistsSystem\\HotelReceptionistsSystem\\HeavensDoor.rpt");


    reportDocument.SetParameterValue("StartDate", startDateFormat);
    reportDocument.SetParameterValue("EndDate", endDateFormat);

    tabPage2.Controls.Clear();

    CrystalReportViewer crystalReportViewer = new CrystalReportViewer();
    crystalReportViewer.Dock = DockStyle.Fill;
    crystalReportViewer.ReportSource = reportDocument;

    tabPage2.Controls.Add(crystalReportViewer);

    tabControl1.SelectedTab = tabPage2;
}
```

Kode diatas adalah kode untuk filter pada menu reports dimana filter tersebut berfungsi sebagai mengatur data yang ingin di cetak dalam crystal report. Filter tersebut didasari oleh waktu check-in dan waktu check-out dimana staff hotel akan menginput tanggal startDate sebagai batasan waktu check-in dan endDate sebagai batasan waktu check-out untuk mencetak laporan. Setelah tanggal dipilih melalui datetimepicker, value nya akan diubah menjadi bentuk string dengan format seperti pada gambar. Lalu ketika button checkout di klik program akan secara otomatis menciptakan report document dengan menggunakan crystalreportviewer pada tabpage ke 2 dengan menampilkan laporan dari HeavensDoor.rpt yang sudah di filterisasi.

2.7.8. Form Child Memo Pad

```
public partial class FormChild : Form
{

    1 reference
    public FormChild()
    {
        InitializeComponent();
    }

    1 reference
    private void FormChild_Load(object sender, EventArgs e)
    {
        TextBox textBox = new TextBox();
        textBox.ReadOnly = false;
        textBox.Font = new Font("Century Gothic", 10);
        textBox.Size = new Size(597, 643);
        textBox.Multiline = true;

        this.Controls.Add(textBox);

    }
}
```

Kode diatas adalah isi dari form child yang akan dipanggil pada menu dashboard. Isi dari kode tersebut hanyalah memunculkan sebuah textbox dan mengatur font dan sizenya serta membuat pengaturan textbox menjadi *multiline.*

Form Child Guest Wishlist (New)

```
public partial class FormChild2 : Form
{
    1 reference
    public FormChild2()
    {
        InitializeComponent();
    }

    1 reference
    private void button1_Click(object sender, EventArgs e)
    {
        List<string> textBoxContents = new List<string>();
        textBoxContents.Add(textBox5.Text);
        textBoxContents.Add(textBox1.Text);
        textBoxContents.Add(textBox2.Text);
        textBoxContents.Add(textBox3.Text);
        textBoxContents.Add(textBox4.Text);

        string filePath = "C:\\Users\\User\\Documents\\Visual Studio 2022\\HotelReceptionistsSystem\\HotelReceptionistsSystem\\Guest Wishlist.txt";

        string content = string.Join(", ", textBoxContents);

        File.AppendAllText(filePath, content + Environment.NewLine + Environment.NewLine);

        MessageBox.Show("Wishlist Successfully Added.");

    }
```

Kode tersebut berasal dari form child Guest Wishlist pada menu *new* dimana fungsinya adalah untuk menginput value-value pada textbox agar dioper ke dalam file txt sesuai dengan path diatas dengan format “, “ pada setiap valuenya.

Form Child Guest Wishlist (Open)

```
4 references
public partial class FormChildWishlist : Form
{
    1 reference
    public FormChildWishlist()
    {
        InitializeComponent();
    }

    1 reference
    private void FormChildWishlist_Load(object sender, EventArgs e)
    {
        string filePath = "C:\\Users\\User\\Documents\\Visual Studio 2022\\HotelReceptionistsSystem\\HotelReceptionistsSystem\\Guest Wishlist.txt";

        string[] fileLines = File.ReadAllLines(filePath);

        foreach (string line in fileLines)
        {

            textBox1.AppendText(line + Environment.NewLine);
        }
    }
```

Kode diatas berfungsi untuk membuka file txt Guest Wishlist.txt dan menampilkan seluruh value yang ada di dalamnya ke dalam textbox pada child form FormChildWishlist.

Form Child Room (Open)

```
public partial class FormChildRoom : Form
{
    1 reference
    public FormChildRoom()
    {
        InitializeComponent();
    }
    private string filePath1 = "C:\\Users\\User\\Documents\\Visual Studio 2022\\HotelReceptionistsSystem\\HotelReceptionistsSystem\\RoomType.txt";
    private string filePath2 = "C:\\Users\\User\\Documents\\Visual Studio 2022\\HotelReceptionistsSystem\\HotelReceptionistsSystem\\RoomPrice.txt";
    private string filePath3 = "C:\\Users\\User\\Documents\\Visual Studio 2022\\HotelReceptionistsSystem\\HotelReceptionistsSystem\\RoomDetail.txt";

    1 reference
    private void textBox1_TextChanged(object sender, EventArgs e)
    {

    }
```

Kode diatas merupakan bagian dari form child room dimana fungsinya adalah untuk membuat private string untuk 3 filepath yang berbeda agar filepath tersebut bisa dipanggil nantinya dengan menggunakan variabel yang diciptakan.

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    textBox1.ReadOnly= false;
    textBox2.ReadOnly= false;
    textBox3.ReadOnly= false;
}

1 reference
private void FormChildRoom_Load(object sender, EventArgs e)
{
    if (File.Exists(filePath1))
    {
        string fileContent1 = File.ReadAllText(filePath1);
        textBox1.Text = fileContent1;
    }

    if (File.Exists(filePath2))
    {
        string fileContent2 = File.ReadAllText(filePath2);
        textBox2.Text = fileContent2;
    }

    if (File.Exists(filePath3))
    {
        string fileContent3 = File.ReadAllText(filePath3);
        textBox3.Text = fileContent3;
    }
}
```

Kode diatas akan berfungsi jika button “edit” pada child form room di tekan dan mengubah property ReadOnly pada 3 textbox yang semula adalah true menjadi false agar data di dalam ketiga textbox dapat diedit oleh user. Dan ketika FormChildRoom ter load, ketiga textbox tersebut akan menampilkan data yang berasal dari 3 file txt yang berbeda.

```
1 reference
private void button2_Click(object sender, EventArgs e)
{
    string fileContent1 = textBox1.Text;
    File.WriteAllText(filePath1, fileContent1);
    string fileContent2 = textBox2.Text;
    File.WriteAllText(filePath2, fileContent2);
    string fileContent3 = textBox3.Text;
    File.WriteAllText(filePath3, fileContent3);
    MessageBox.Show("Room Detail Updated");

    textBox1.ReadOnly = true;
    textBox2.ReadOnly = true;
    textBox3.ReadOnly = true;
}
```

Setelah user/staff hotel melakukan edit pada textbox tersebut, ketika user mengklik tombol update maka semua value yang terdapat di dalam masing-masing textbox akan di update

ke dalam 3 file txt tersebut dan status ReadOnly dari 3 textbox tersebut akan dikembalikan menjadi true.

BAB III

KESIMPULAN

3.1 Kesimpulan

Aplikasi dekstop yang kami buat memiliki tujuan dalam meningkatkan efektivitas dan efisiensi dalam manajemen hotel ini. Aplikasi hotel ini dapat membantu mengelola reservasi hotel, ketersediaan kamar, pengelolaan keuangan, analisis data yang bisa dijadikan bahan evaluasi oleh hotel, dan membantu meningkatkan keamanan data yang bersifat sensitif. Dengan adanya aplikasi dekstop ini, dapat membantu hotel dalam operasional yang lebih mudah diatur sehingga dapat memudahkan staf hotel dalam mengelola data.

Peran Anggota

A. Deskripsi peran dan tanggung jawab masing-masing anggota

Nathan Villbert Kosasih : Merancang dan membuat aplikasi pada visual studio 2022, membuat database, menghubungkannya database pada aplikasi, membuat penjelasan mengenai codingan.

Fernando Yulius Hadiarto : Membuat Laporan (Latar Belakang, Tujuan, Manfaat, Cara kerja proses bisnis, check in & check out), dan membuat logo

Ongko Fillberto : Merancang dan menyusun Entity Relationship Diagram, membantu pembuatan ppt

Hans Samuel Gunawan : Mengisi laporan (Cara kerja aplikasi, dan flowchart proses bisnis)

Felix Samuel Leo : Mengisi laporan (Menu/fitur aplikasi dan teknologi .net), membuat ppt