**UAS PENGEMBANGAN PIRANTI LUNAK**

**DOKUMENTASI *PROJECT***

**“PERJALANAN PENERBANGAN”**

**ANGGOTA KELOMPOK:**

**ALBERT CENDERAWAN (03082190015)**

**HUBERT DANIEL RUSLI (03082190024)**

**DAFTAR ISI**

1. **BAB I PENDAHULUAN**
   1. LATAR BELAKANG
   2. RUMUSAN MASALAH
   3. BATASAN MASALAH
   4. TUJUAN MASALAH
   5. MANFAAT PENELITIAN
   6. METODOLOGI PENELITIAN
2. **BAB II DATA TEORI**
   1. DEFINISI DATABASE
   2. DEFINISI USE CASE DIAGRAM
   3. DEFINISI SEQUENCE DIAGRAM
   4. DEFINISI DAN PENGGUNAAN *SOURCE CODE*
   5. DEFINISI DAN PENGGUNAAN IDE
3. **BAB 3 PEMODELAN DIAGRAM**
   1. USE CASE DIAGRAM
   2. SEQUENCE DIAGRAM
4. **BAB 4 PEMODELAN TAMPILAN APLIKASI**
   1. PEMBUKTIAN DAN HASIL DARI APLIKASI
5. **BAB 5 PENUTUP**
   1. KESIMPULAN
   2. SARAN
6. **DAFTAR PUSTAKA**

**BAB I**

**PENDAHULUAN**

1. **LATAR BELAKANG**

Saat ini, kita berada dalam zaman teknologi yang berpacu dengan waktu yang begitu cepat. Banyak aktivitas dan tugas yang harus ditangani dalam waktu yang cepat maupun bersamaan. Oleh sebab itulah, teknologi dibutuhkan agar dapat memudahkan kerja kita.

Dengan adanya adaptasi dengan teknologi, kita dapat menghemat banyak waktu dan mengurangi risiko *human-error* dalam kegiatan kita dan dengan memberikan data kepada orang lain. Tentunya kegiatan kita yang sibuk bisa dimudahkan berkat pertolongan teknologi. Itulah sebabnya mengapa dunia bisnis membutuhkan teknologi, baik untuk skala kecil maupun skala besar.

Salah satu sektor bisnis yang sangat membutuhkan teknologi adalah sektor *Travel* (perjalanan udara, darat, maupun laut). Saat ini, permintaan konsumen terhadap hotel, perjalanan darat, dsb. sangat tinggi karena meningkatnya laju mobilitas konsumen. Tidak hanya itu, budaya/ hobi *travelling* sudah semakin luas pada kaum milenial zaman sekarang. Oleh sebab itulah, pihak hotel, penyedia perjalanan, dll. harus bersedia meningkatkan pelayanan terhadap konsumen tersebut. Banyak orang suka *travelling* tanpa memakai *guide tour* maka dari pihak hotel, perjalanan, dan lain lain mereka harus memberikan informasi kepada mereka dengan menggunakan teknologi sekarang yaitu *website* maupun *mobile app*. Salah satu di antaranya ialah dalam bidang perjalanan penerbangan.

Bagi para wisatawan yang sering pergi keluar kota dan/ atau antarprovinsi, pesawat menjadi moda transportasi terbaik dari semua pilihan. Hal ini karena waktu perjalanan yang sangat cepat, pelayanan yang lebih baik, keamanan lebih, dan tentunya kita bisa rileks dalam pesawat sepanjang perjalanan.

Akan tetapi, perjalanan pesawat juga memiliki tantangan. Masalah pertama adalah pada saat *Check In*. Umumnya, bandara bisa sangat ramai, terutama di saat hari libur. Hal ini karena arus mobilitas yang cukup tinggi, sehingga antrean bisa mengular di pintu masuk, loket, maupun gerbang *Check In*.

Selain itu, jadwal penerbangan sudah bersifat final. Oleh karena itu, sekalipun kita ingin penerbangan pada pagi hari, kalau jadwalnya sudah siang, ya, kita harus ikut penerbangan siang. Alternatif pertama ialah dengan mengganti maskapai penerbangan apa adanya. Selain itu, alternatif lainnya adalah dengan memilih penerbangan dengan transit. Misalnya penerbangan Medan (KNO) - Bandung (BDO), tapi transit di Jakarta (CGK).

Masalah selanjutnya adalah di bidang kenyamanan (*hospitality*). Meski sudah maju, teknologi sekarang masih belum dapat memberikan kenyamanan yang cukup baik karena teknologi sekarang masih belum bisa membuat sebuah teknologi yang bisa menggunakan *human feeling*.

Masalah ketiga yang dialami konsumen adalah di bidang pembiayaan. Saat ini, kita pasti punya minimal 1 (satu) kartu debit/ kredit. Selain itu, kita juga memiliki minimal 1 (satu) buah *virtual account*, seperti via OVO, GoPay, Dana, LinkAja, dll. Oleh karena itulah, konsumen menginginkan pembayaran secara *online* dengan aman dan nyaman.

1. **RUMUSAN MASALAH**
2. Bagaimana cara aplikasi untuk menangani pembayaran yang dilakukan konsumen secara *online*?
3. Bagaimana bentuk fitur dari aplikasi yang dapat menjamin *human feeling* konsumen dalam memesan tiket?
4. Bagaimana caranya agar tidak terjadi masalah *double payment* dalam transaksi melalui aplikasi tersebut?
5. Bagaimana cara aplikasi mengontrol & mengatur setiap akun yang sudah terdaftar dalam aplikasi tersebut?
6. **BATASAN MASALAH**

Mengingat keterbatasan waktu dan sumber daya yang tersedia dalam proyek ini, kami membatasi dan memfokuskan masalah hingga mencapai batasan - batasan sebagai berikut:

1. Sektor “*Tour and Travel*” yang dibahas dalam aplikasi hanya dalam lingkup “Perjalanan Udara”, seperti perjalanan pesawat terbang.
2. Fitur pembayaran terbatas hanya untuk 2 jenis, misalnya melalui *Transfer* (melalui *Virtual Account*) maupun Tunai.
3. Fitur pelayanan dan penerimaan keluhan konsumen masih terbatas dan sifatnya belum *real-time*.
4. Basis data (*database*) yang digunakan masih bersifat *offline*, namun memiliki cara kerja yang mirip dengan prinsip pengaksesan dan penyimpanan data di *database* secara *online*.
5. Aplikasi ini masih dalam tahap uji coba dan masih banyak fitur yang belum tersedia, sehingga perlu pengembangan dan evaluasi lebih lanjut.
6. **TUJUAN PENELITIAN**

Tujuan penelitian ini adalah untuk menganalisis dan merancang aplikasi pembelian tiket pesawat terbang secara *online*. Selain itu, aplikasi ini juga bertujuan agar para *programmer* lainnya dapat mengenal dan memahami algoritma sederhana dari aplikasi tersebut.

1. **MANFAAT PENELITIAN**

Manfaat yang diharapkan dalam penelitian ini adalah agar:

1. Konsumen dapat dipermudah dalam melaksanakan pembelian tiket secara *online*.
2. Pemilik usaha di bidang “*Travel*” dapat merasakan kemudahan karena dapat menerima dan mengelola data konsumen dengan aman dan nyaman.
3. Pengelola bandara dapat merasakan kemudahan dan fleksibilitas, karena pemesanan dapat dilakukan secara *online* sehingga dapat memotong jumlah antrean di loket bandara.
4. **METODOLOGI PENELITIAN**

Dalam penelitian, kami menggunakan metode pengembangan aplikasi berupa *Waterfall Method*, dengan melalui langkah - langkah sebagai berikut:

1. **STUDI LITERATUR**

Dalam tahap ini, kami mencari informasi penting yang didapat dari *website* perjalanan dan pemesanan tiket secara *online*, seperti Traveloka.com dan Tiket.com. Selain untuk memahami kinerja *website* tersebut secara umum, kegiatan akses *website* tersebut juga berguna agar dapat memahami tampilan *website* mana yang paling ideal agar *user* dapat memesan tiket dengan nyaman.

1. **TAHAP *REQUIREMENT***

Dalam tahap ini, kami mengakses beberapa situs penerbangan resmi, seperti situs jadwal penerbangan di Bandara Internasional Kualanamu, Medan. Dengan demikian, kami dapat memantau kegiatan penerbangan apa saja yang ditemukan dalam bandara tersebut.

1. **TAHAP ANALISIS**

Dalam tahap ini, kami menyusun data dan mengorganisasikannya ke dalam bentuk model diagram, seperti melalui *Use Case Diagram* dan *Sequence Diagram*. Dari semua *software* yang ada, kami memilih aplikasi *StarUML* karena fitur pembuatan diagram yang sudah jadi, sehingga mudah digunakan.

1. **TAHAP *DESIGN***

Dalam tahap ini, kami menyusun dan merancang sistem yang dibutuhkan untuk menyimpan dan mengelola data dengan baik. Salah satunya ialah dengan pengembangan *database* dengan menggunakan *Microsoft Access*.

1. **TAHAP *DEVELOP***

Dalam tahap ini, kami menyusun dan membangun program dengan menggunakan bahasa pemrograman C#. Selain itu, kami juga menggunakan IDE berupa *Visual Studio*.

1. **TAHAP *TESTING***

Dalam tahap ini, sistem dalam aplikasi diuji coba dengan menggunakan *partial testing*, dengan metode *Blackbox Testing*. Hal ini bertujuan agar dapat memahami dan memastikan apakah sistem dalam aplikasi tersebut sudah berjalan lancar dan sesuai atau belum.

1. **TAHAP EVALUASI**

Dalam tahap ini, kami merespon keluhan pengguna, apakah aplikasi kami mengalami *error*, terjadi kesalahan *output*, dsb. sehingga kami dapat meng-*update* aplikasi ini menjadi lebih sesuai dengan keinginan pengguna.

1. **TAHAP PEMBUATAN LAPORAN DOKUMENTASI**

Sistematika proses dan hasil penelitian dirangkum dan dicantumkan dalam laporan akhir pengembangan aplikasi, yang akan dipresentasikan ke dosen.

1. **SISTEMATIKA PENULISAN**

Bab I (Pendahuluan)

Berisi latar belakang, rumusan dari masalah yang dibahas, batasan - batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

Bab II (Data Teori)

Berisi tentang teori yang digunakan untuk memperkuat pendapat maupun pandangan kami dalam penyusunan laporan pengembangan aplikasi.

Bab III (Permodelan Diagram)

Berisi proses analisa dan perancangan *project* secara visual, yakni menggunakan diagram.

Bab IV (Permodelan Tampilan Aplikasi)

Berisi evaluasi sistem yang dirancang dari cara penggunaan sistem. Di bagian ini, kami menampilkan bentuk tampilan aplikasi, mulai dari awal hingga akhir penggunaan sistem.

Bab V (Penutup)

Berisi kesimpulan dari hasil penelitian, serta saran dalam penelitian ini, sebagai jejak untuk pengembangan - pengembangan aplikasi ini selanjutnya.

**BAB II**

**DATA TEORI**

1. **DEFINISI *DATABASE***

*Database* merupakan sebuah kumpulan informasi atau data yang disimpan di dalam komputer (*offline*) maupun dari internet (*online*) yang lebih dikenal dengan *cloud database*. *Database* lalu diperiksa menggunakan suatu program komputer untuk memperoleh sebuah informasi dari basis data tersebut. Informasi tersebut diambil dan diolah secara otomatis, sesuai dengan keinginan pengguna. *Database* yang baik memiliki ciri - ciri sebagai berikut:

* Memiliki data yang akurat dan dapat dikelola dengan baik
* Memiliki kenerja yang sesuai dan dapat diterima
* Memiliki data yang dapat dibaca dan diterima oleh *user*
* Memiliki ruang penyimpanan data yang cukup, dsb.

Dalam *project* Pengembangan Piranti Lunak ini, kami memilih Microsoft Office Access sebagai *database* pilihan kami. Selain karena pengaturannya yang cukup mudah, *database* ini juga dapat terkoneksi secara baik dan lancar dengan bahasa pemrograman C# dalam *project* ini.

1. **DEFINISI *USE CASE DIAGRAM***

*Use case diagram* adalah sebuah kegiatan maupun interaksi yang saling berkesinambungan antara aktor dan juga sistem. Dengan kata lain, *use case diagram* bisa digunakan untuk mengembangan *software* atau sistem informasi dengan lebih singkat dan juga mudah dibaca oleh sesama rekan tim. Dalam *use case diagram*, terdapat elemen (simbol) penting di antaranya:

* *Actor*

Merupakan pihak (orang/ sistem) yang berinteraksi dengan sistem informasi tersebut. Inilah elemen yang paling utama dalam *use case diagram*. Tanpa adanya *actor*, tidak akan ada *use case diagram*.

* *Use Case*

Merupakan kegiatan/ aktivitas yang dilakukan oleh *actor*. *Use case* berbentuk seperti sebuah unit yang saling berelasi, baik dengan *actor* maupun dengan *use case* yang lain.

* *Association*

Merupakan garis komunikasi antara *actor* dengan *use case*. Garis ini menandai aktivitas yang dilakukan *actor* dalam sistem.

* Panah *extend*

Merupakan garis panah putus - putus yang menghubungkan *use case* dengan *use case*. Garis ini menandai bahwa *use case* tersebut tidak harus dikerjakan (sifatnya *optional*). Panah ini menuju ke *use case* utama maupun tambahan.

* Panah *include*

Merupakan garis panah putus - putus yang menghubungkan *use case* dengan *use case*. Garis ini menandai bahwa *use case* tersebut wajib dikerjakan (sifatnya *mandatory*). Panah ini menuju ke *use case* tambahan.

* *Generalization*

Merupakan garis panah yang menghubungkan *use case* dengan *use case*. Garis ini menandai hubungan generalisasi-spesialisasi antara *use case* yang satu dengan yang lain (prinsip *Inheritance*). Panah ini menuju ke *use case* utama (umum).

1. **DEFINISI *SEQUENCE DIAGRAM***

*Sequence diagram* adalah sebuah diagram yang menjelaskan skema aktivitas yang disusun secara terstruktur dan berurutan (*sequential*). Berdasarkan informasi dari beberapa sumber, *sequence diagram* menjelaskan bagaimana sebuah operasi dilakukan, pesan (*message*) apa yang dikirim, serta kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu, tapi sebenarnya lebih diatur berdasarkan urutan kronologis kejadian (kejadian yang satu tidak bisa berjalan tanpa kejadian sebelumnya). Dalam *sequence diagram*, terdapat elemen - elemen utama sebagai berikut:

* Aktor/ class

Merupakan pihak (*object*) yang terlibat dalam sistem.

* Activation Boxes

Merupakan sebuah acuan dimana objek/ aktor mulai berpartisipasi dalam sistem. Partisipasi ini dapat berupa pengiriman pesan maupun penerimaan pesan.

* Lifeline

Merupakan garis putus - putus vertikal yang bertugas sebagai

* Message

Merupakan pesan yang diberikan antara *object (actor)* yang satu dengan yang lain. Dengan *message*, *object* bisa saja meneruskan *message* tersebut ke *actor* yang lain maupun memberi pesan balasan.

Melalui deskripsi visual tentang urutan dan pola aktivitas tersebut, kita dapat memahami:

1. Pihak manakah yang sangat berperan aktif dalam sistem tersebut?
2. Apakah salah satu pihak berperan sedikit atau tidak?
3. Apa saja & berapa banyak aktivitas yang melibatkan 2 pihak yang bersangkutan?
4. Berapa banyak aktivitas yang berlangsung dalam sistem tersebut?
5. **DEFINISI DAN PENGGUNAAN *SOURCE CODE***

*Sourcecode* adalah kumpulan baris berisi teks berbahasa pemrograman yang digunakan pada sebuah komputer. *Sourcecode* dapat berupa perintah, pernyataan, maupun deklarasi yang dapat dipahami oleh 2 pihak sekaligus (pemrogram dan komputer).

Secara umum, *sourcecode* merupakan hal yang sangat penting dalam pemrograman, karena di sinilah “karya/ inspirasi/ komunikasi” pengembang/ pemrogram divisualisasikan. *Sourcecode* biasanya ditulis dengan posisi yang rapi maupun terstruktur, tergantung dari bagaimana regulasi penulisan *syntax* tersebut. *Sourcecode* yang bagus tidak hanya dilihat dari kerapiannya, melainkan dari seberapa efektif dan efisien *sourcecode* tersebut bagi pihak yang lain, seperti memori.

Dalam *project* kali ini, kami menggunakan *source code* dengan bahasa pemrograman C#, karena bahasa ini bersifat umum (dapat digunakan di banyak *tools*), berorientasi objek (OOP), serta menggunakan bahasa yang cukup modern. Selain itu, kami juga terbiasa dengan bahasa pemrograman ini, mengingat bahasa C# juga digunakan pada mata kuliah sebelumnya.

1. **DEFINISI DAN PENGGUNAAN IDE**

IDE (*Integrated Development Environment*) adalah *tool* pemrograman yang dapat berfungsi sebagai *text editor, code library, compiler,* maupun *tools* untuk melakukan *testing* program. Dengan demikian, IDE ibarat satu *software* yang sifatnya *All-In-One*. IDE juga dapat diakses secara *online* maupun secara *offline*, dengan tampilan yang berbeda pula. Akan tetapi, semua IDE umumnya memiliki fitur sebagai berikut:

* Fitur *text editor*

Di sinilah, kita bisa menginput *code* kita dengan rapi dan teratur. IDE yang canggih biasanya punya keahlian untuk merapikan *syntax* secara otomatis.

* Fitur pembeda *syntax*

Dalam melakukan pemrograman, kita pasti tidak asing dengan warna - warni dalam *text editor* kita. Itulah fitur pembeda *syntax* tersebut. Dengan demikian, efek *typo* dapat diminimalisir

* Fitur *code completion*

Fitur ini mirip dengan fitur *autocorrect* dalam *keyboard* *smartphone* kita. Dengan fitur ini, kita tidak perlu susah payah mengetik dan mengecek *code* yang kita tulis.

* Fitur menjalankan (*running*) *code*

Dengan fitur ini, *programmer* tidak perlu susah payah memasang aplikasi lain untuk menjalankan *code* kita.

* Fitur *debugging*

Dengan fitur ini, setiap karakter dalam *code* kita dicek secara otomatis, sehingga kita mudah mendapatkan *error* ketika sedang *compile* *code* kita. Bahkan, ada IDE yang langsung memberi pesan petunjuk (*hint*) secara jelas, terkait mana variabel yang salah, mana *function* yang kurang tepat, dll.

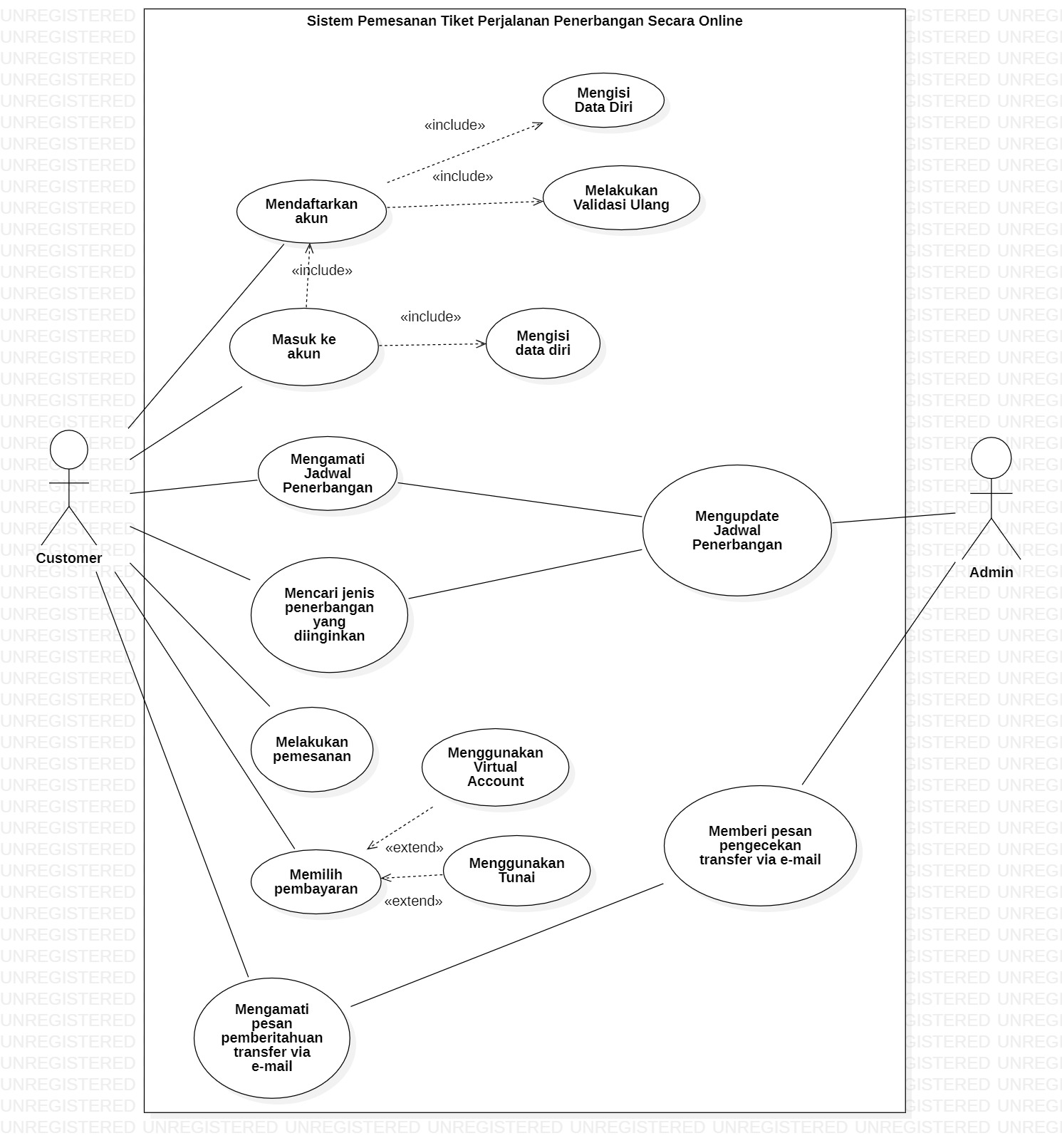
Dalam *project* ini, kami menggunakan salah satu IDE yang terkenal, yaitu **Visual Studio** dari Microsoft. Meskipun IDE ini cukup berat dalam memori, namun fitur - fitur di dalamnya sangat lengkap, sehingga memudahkan kami dalam menyusun *form* dan melakukan pemrograman.

**BAB III**

**PEMODELAN DIAGRAM**

1. **USE CASE DIAGRAM**

Kami menggunakan *offline tool* bernama StarUML untuk membangun *Use Case Diagram* ini. Diagram ini terdiri dari 2 (dua) buah aktor, yakni *Customer* dan *Admin*. *Customer* berperan seperti halnya *user* ketika menjelajahi program tersebut. *Admin* merupakan pihak yang bertanggung jawab atas pelayanan program dalam berinteraksi dengan *user*. Berikut ini adalah tampilan *Use Case Diagram* tersebut:

****

1. **SEQUENCE DIAGRAM**

Kami menggunakan *offline tool* bernama StarUML untuk membangun *Sequence Diagram* ini. Diagram ini memiliki 2 (dua) *class*, yaitu *customer*, dan *Perusahaan*. Berikut adalah tampilan *Sequence Diagram* dalam projek Pengembangan Piranti Lunak ini:



****

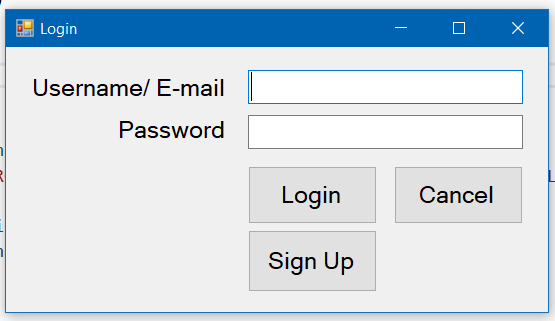
****

**BAB IV**

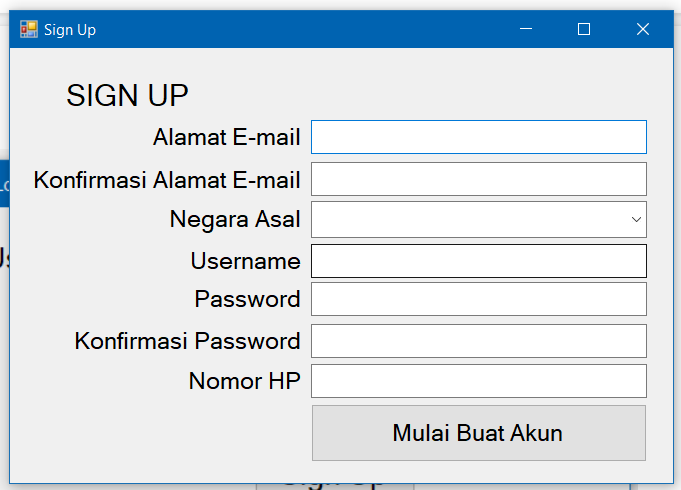
**PEMODELAN TAMPILAN APLIKASI**

**4.1 PEMBUKTIAN DAN HASIL DARI APLIKASI**

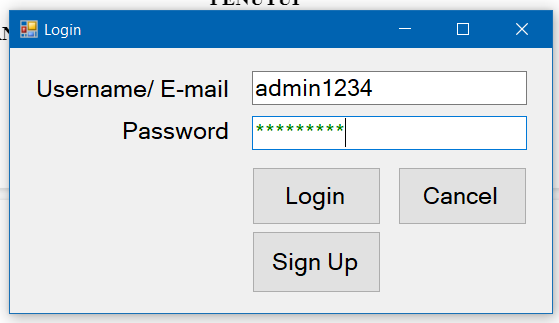
Berikut ini adalah tampilan *form* aplikasi yang dijalankan dari awal hingga akhir:



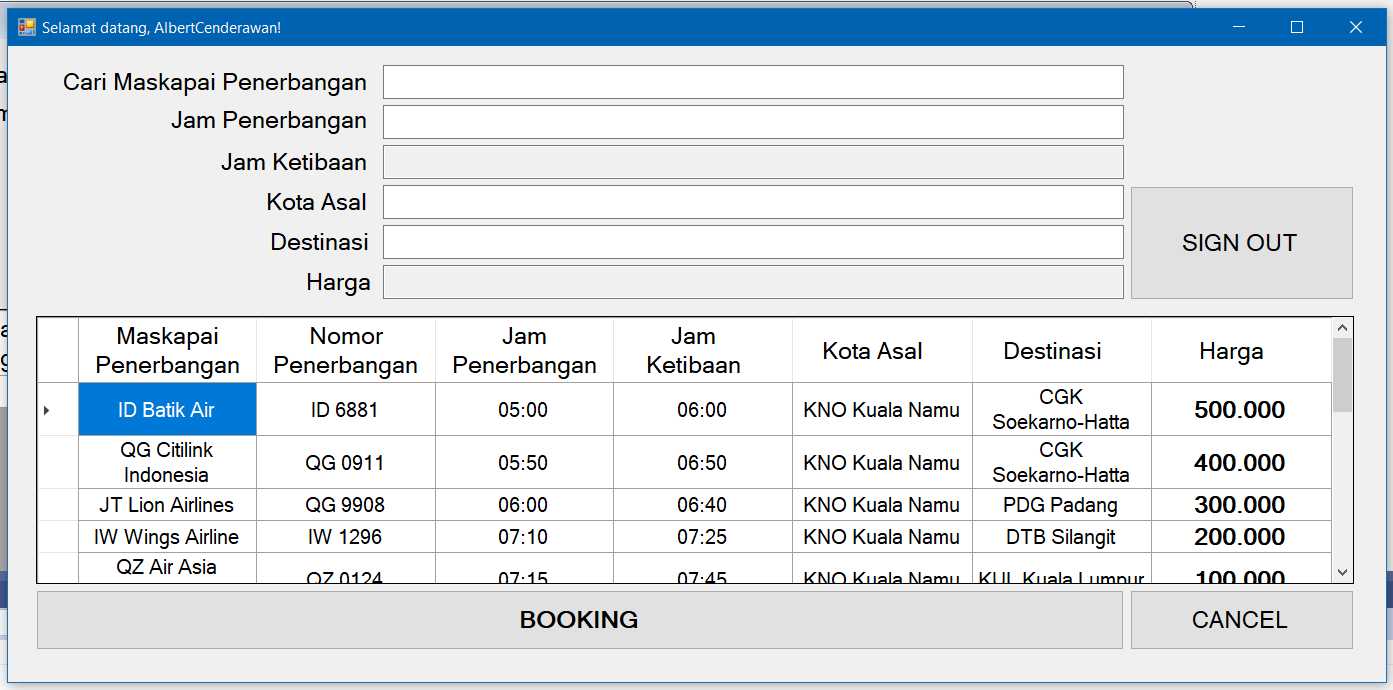
Ini adalah tampilan form *Login* setelah *user* menginput *username* dan *password*. Jika *user* belum mendaftarkan akun sama sekali, *user* harus menekan tombol *Sign Up* untuk menambah akun tersebut.



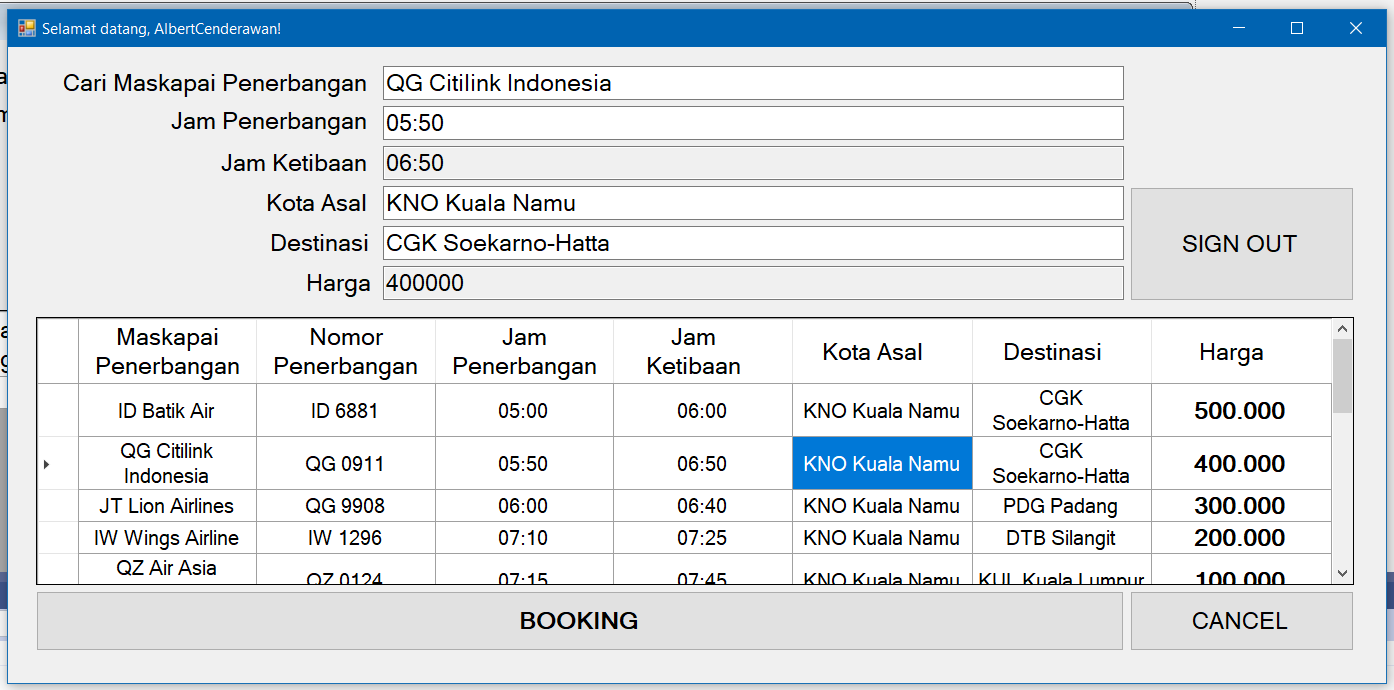
Setelah mengisi data yang tanyakan kepada *user* maka *user* tinggal menekan tombol “Mulai Buat Akun”. maka dari data tersebut *user* sudah mendapatkan akun untuk masuk ke dalam aplikasi tersebut. Setelah itu, program akan mengarahkan *user* ke dalam form *Login* .



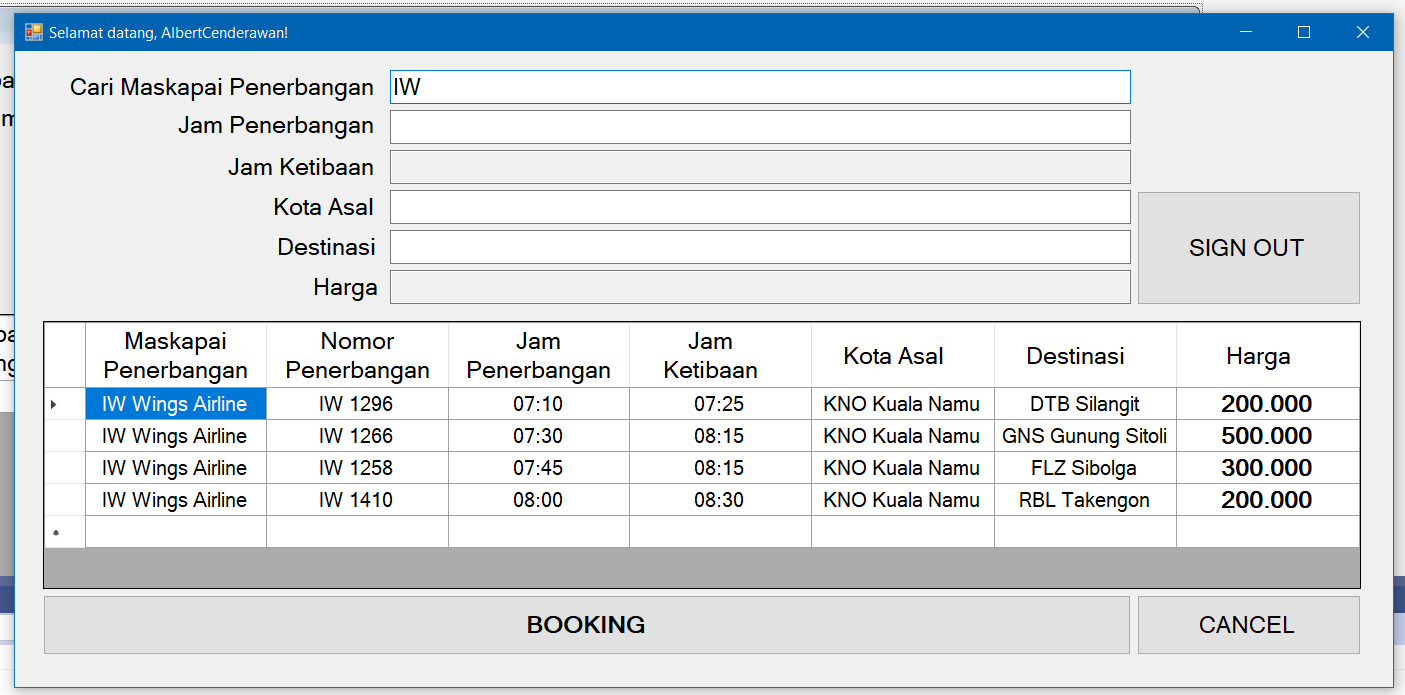
Setelah *Login*, *user* dapat mengakses Jadwal Penerbangan, dengan tampilan *form* seperti di bawah ini. Dari sini, terlihat jelas data - data jadwal penerbangan yang dituliskan di dalam *data grid view* dengan tampilan sebagai berikut.



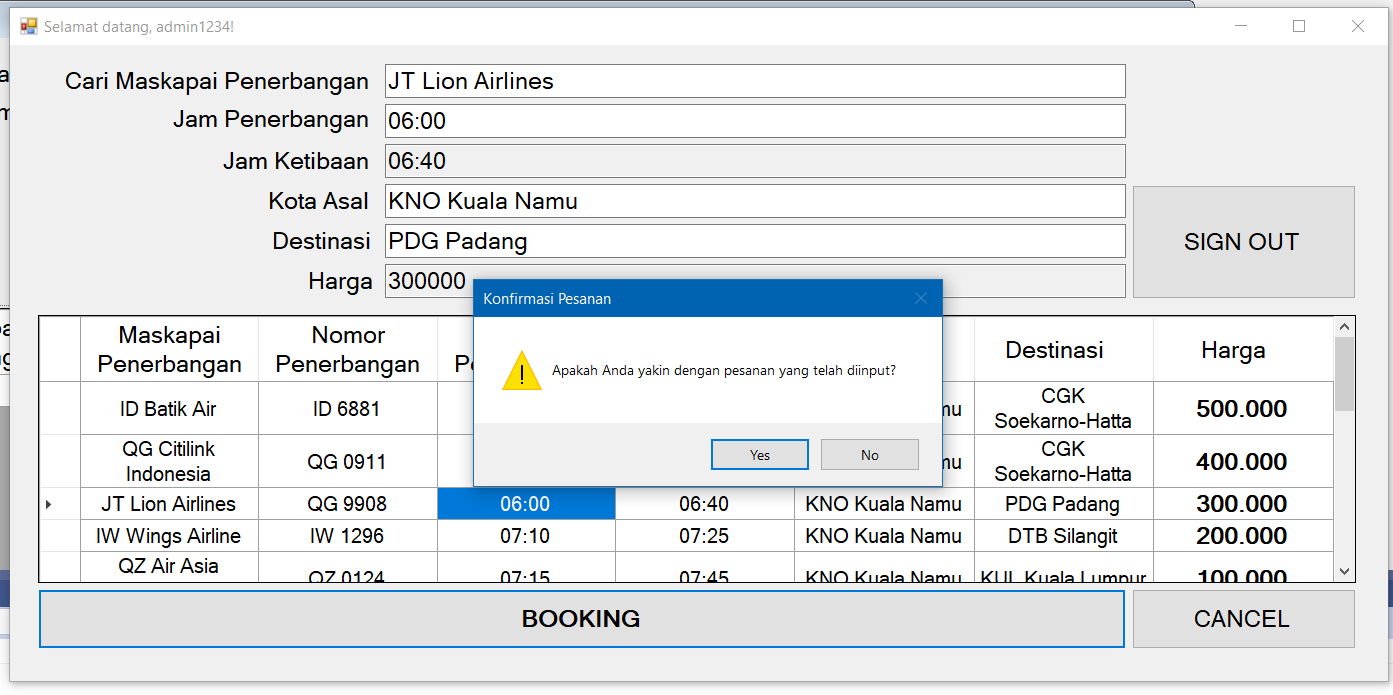
*User* dapat mengisi nama maskapai penerbangan yang ingin dicari, jam penerbangan, kota asal, destinasi, serta tanggal untuk penerbangan, sesuai dengan yang diinginkan oleh *user*. Untuk memilih jenis dan nomor penerbangan yang diinginkan, *user* bisa langsung mengklik *cell* yang terkait. Dengan demikian, data dalam *record* ditampilkan dalam *TextBox.*



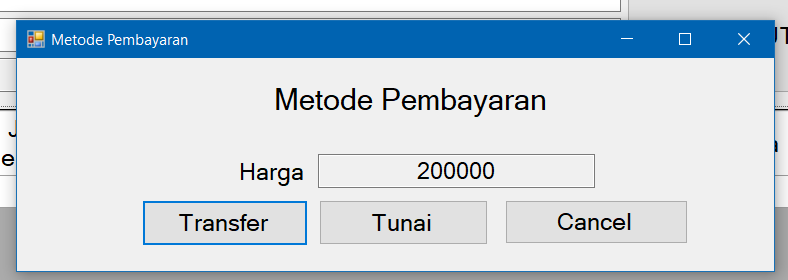
Cara lainnya ialah dengan menggunakan fitur *TextBox*, dimana *user* hanyamengisi beberapa karakter teks, lalu tekan Enter. Hasil yang ditampilkan adalah sebagai berikut:



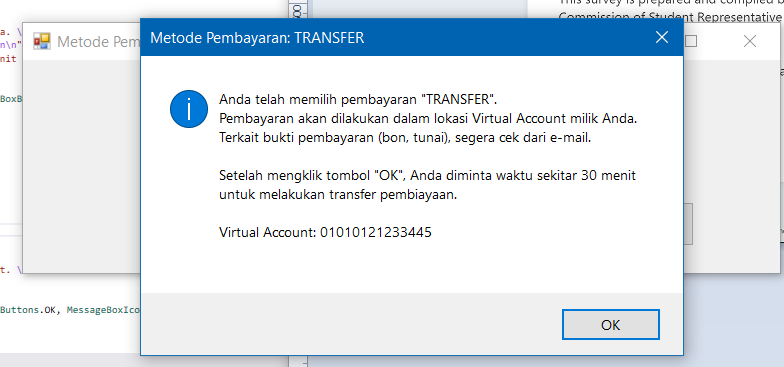
Contoh di atas merupakan pencarian maskapai penerbangan “IW Wings Airline”, dimana *user* hanya menginput “IW” saja. Setelah *user* memilih dan ingin memesan/ membeli tiket penerbangan yang diinginkan, klik tombol “*Booking*”. Pesan konfirmasi akan keluar, untuk meyakinkan *user*, apakah sudah yakin atas pemesanan itu atau tidak.



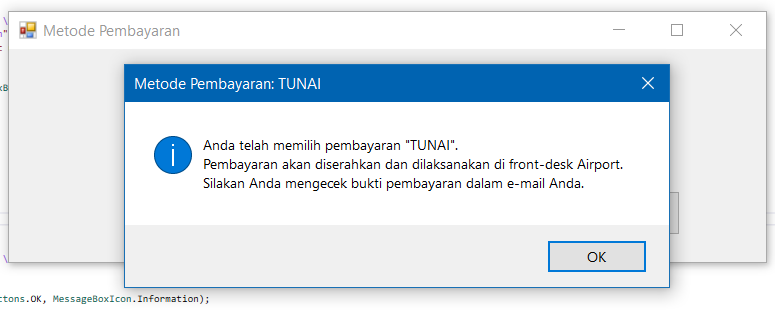
Setelah itu, *user* diarahkan ke *form* Pembayaran. Form ini memberi pilihan kepada *user* untuk memilih, apakah berupa “*Transfer*” atau “Tunai”. Apabila *user* telah berpikir ulang untuk membatalkan pembelian, *user* dapat memilih tombol “*Cancel*”.



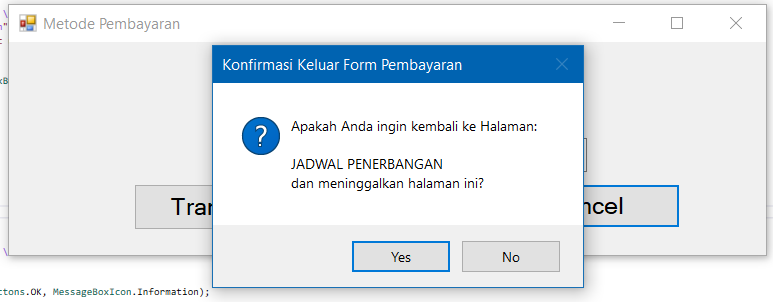
Jika menggunakan metode “Transfer”, maka akan muncul kotak MessageBox seperti dalam tampilan di bawah ini. Setelah *user* mengklik “OK”, *user* langsung diarahkan ke form *Login* untuk menghindari adanya *double payment*.



Jika menggunakan metode “Tunai”, maka akan muncul kotak MessageBox seperti dalam tampilan di bawah ini. Setelah *user* mengklik “OK”, *user* langsung diarahkan ke form *Login* untuk menghindari adanya *double payment*.



Jika *user* mengklik tombol “*Cancel*”, hal ini berarti *user* telah melakukan pembatalan terhadap pemesanan. Untuk menghindari hal tersebut, kotak *MessageBox* akan keluar dengan pesan sebagai berikut:

Apabila *user* menjawab “*Yes*”, *user* akan diarahkan oleh program ke *form* Jadwal Penerbangan kembali. Demikian seterusnya sampai akhirnya *user* meninggalkan aplikasi tersebut.

**BAB V**

**PENUTUP**

1. **KESIMPULAN**

Saat ini, banyak orang (terutama generasi muda) yang hobi *Travelling*, sehingga mobilitasnya tinggi. Oleh sebab itulah, penyedia perjalanan memanfaatkan kesempatan ini untuk mempromosikan penerbangan dan memberi kenyamanan lebih, terutama pada hari libur dan akhir pekan. Salah satunya ialah dalam perjalanan penerbangan.

Dalam *project* ini, kami memilih topik “Perjalanan Penerbangan”. Alasan kami memilih aplikasi dengan topik tersebut adalah agar dapat mempermudah para konsumen dalam melakukan pengecekan jadwal dan pemesanan (*booking*) tiket penerbangan yang tersedia dalam penerbangan, seperti di Bandara Internasional Kualanamu, Deli Serdang.

Selain itu, dibandingkan dengan moda transportasi yang lain, pesawat terbang dipilih karena menghemat waktu, memberi pelayanan lebih, aman, serta memberi kerileksan lebih semasa perjalanan pesawat. Akan tetapi, perjalanan penerbangan juga mengalami tantangan. Salah satunya ialah pada saat proses *Check In*, jadwal penerbangan, faktor kenyamanan, dan pada metode pembayaran yang diberikan.

Oleh sebab itulah, kami memilih *project* dengan merancang, mendesain, dan membangun aplikasi pengecekan jadwal pesawat terbang yang dilengkapi dengan fitur pembayaran, yang nantinya diarahkan ke e-mail masing - masing. Aplikasi tersebut dilengkapi fitur *Sign In*, dimana data akan disimpan dalam *database* secara langsung.

Dalam *project* ini, kami memilih bahasa pemrograman C# sebagai bahasa utama, dimana semuanya dikerjakan dalam Visual Studio (sebagai IDE untuk *project* tersebut). Terkait dengan penyimpanan data, kami memilih menggunakan Microsoft Access secara *offline* sebagai *database* *project* kami. Meski merupakan *offline database*, tetapi cara kerjanya mirip seperti pada konsep penyimpanan dan pengaksesan data dalam *database* pada umumnya.

1. **SARAN**

Demikian makalah dokumentasi ini kami susun. Apabila ada kesalahan, kami memohon maaf, dan kami akan memperbarui makalah ini untuk *project* bertema Pengembangan Piranti Lunak ke depannya. Kami juga memohon maaf bila terjadi kesalahan penulisan, struktur kata, penyampaian, dll. yang sekiranya mengurangi kenyamanan para pembaca. Kami berharap agar penulisan makalah ini dapat dikembangkan ke arah yang lebih berguna/ lebih baik, sehingga dapat menyenangkan para pembaca pada *project* serupa mendatang.

Selain itu, kami juga menyadari bahwa isi dokumentasi yang kami kerjakan masih jauh dari kata sempurna. Kami juga menyadari bahwa pengerjaan dokumentasi pada *project* kami sedikit terhambat akibat keterbatasan waktu dan tempat. Oleh sebab itu, kami memohon maaf dan bersedia untuk mengubah dokumentasi dan pengerjaan *project* ini ke dalam bentuk yang lebih baik lagi di masa depan.

Akhir kata, sebagai penutup, atas kerjasama dan perhatian semua pembaca, kami selaku kelompok Pengembangan Piranti Lunak mengucapkan terima kasih.

**DAFTAR PUSTAKA**

Ariiyadi, Teguh. (2019, 29 Mei). Tips Menjaga Source Code Program Milik Perusahaan. Diakses pada 6 April 2021, dari <https://www.hukumonline.com/klinik/detail/ulasan/lt52b6be1a30b85/tips-menjaga-isource-code-i-program-milik-perusahaan/#:~:text=Source%20code%20berisi%20sekumpulan%20instruksi,code%20termasuk%20dalam%20program%20komputer>

Arini, Sukma. (2017, Oktober). Apa saja kriteria database yang baik?. Diakses pada 7 April 2021, dari <https://www.dictio.id/t/apa-saja-kriteria-database-yang-baik/12814/2>

Hidayati, Khairina F. (2021, 3 Februari). Integrated Development Environment: Satu Aplikasi untuk Banyak Fungsi Programming. Diakses pada 7 April 2021, dari <https://glints.com/id/lowongan/integrated-development-environment-adalah/#.YG1P9D8xVEY>

MateriDosen. (2017, 2 April), Use Case Diagram, Lengkap Studi Kasus dan Contoh Use Case. Diakses pada 7 April 2021, dari <http://www.materidosen.com/2017/04/use-case-diagram-lengkap-studi-kasus.html>

Prayudita, Raheza. (2018, 22 Juli). Sequence Diagram. Diakses pada 7 April 2021, dari <http://www.materikuliahif-unpas.com/2018/07/sequence-diagram.html>

Sano, Albert Verasius Dian. (n.d.). DIAGRAM SEQUENCE DALAM ANALISA & DESAIN SISTEM INFORMASI. Diakses pada 7 April 2021, dari <https://binus.ac.id/malang/2020/12/diagram-sequence-dalam-analisa-desain-sistem-informasi/>