

BAB 2

TINJAUAN PUSTAKA

2.1 Profil Perusahaan

PT Yudhistira Ghalia Indonesia merupakan sebuah perusahaan yang bergerak dalam percetakan dan penerbitan buku. Buku-buku yang dipasarkan berfokus pada buku mata pelajaran, buku yang diproduksi mulai dari buku SD, SMP, SMA hingga SMK. PT Yudhistira Ghalia Indonesia mempunyai beberapa cabang yang tersebar di seluruh Indonesia, salah satunya cabang kota Bandung. PT Yudhistira cabang Bandung terbagi lagi menjadi dua bagian yaitu PT Yudhistira Ghalia Bandung dan UMK Bandung. Untuk UMK Bandung khusus mendistribusikan produk buku untuk sekolah SMK saja.

2.2 Sejarah Perusahaan

Perusahaan penerbitan yang merupakan bagian dari Ghalia Indonesia Group ini, bernama resmi PT Yudhistira Ghalia Indonesia, didirikan tahun 1971 di Jakarta. Pada mulanya memakai merek Ghalia Indonesia, dengan menerbitkan sebagian besar buku-buku hukum, perundang-undangan, sosial, politik, dan ekonomi di Indonesia. Mulai menerbitkan dan mencetak buku sekolah tahun 1978, memakai merek Yudhistira. Sampai saat ini, setelah berusia lebih dari 45 tahun, penerbitan ini tetap setia melayani kebutuhan bahan ajar bagi siswa dan pendidik di tingkat SD/MI, SMP/MTs, SMA/MA, dan SMK, serta untuk buku-buku teks perguruan tinggi, yang masih tetap memakai merek Ghalia Indonesia.

2.3 Visi dan Misi

Sebuah Perusahaan tentu mempunyai visi dan misi yang mendukung ataupun menjadi ciri khas Perusahaan tersebut. Adapun Visi dan misi dari PT Yudhistira Ghalia Indonesia cabang UMK Bandung sebagai berikut:

2.3.1 Visi

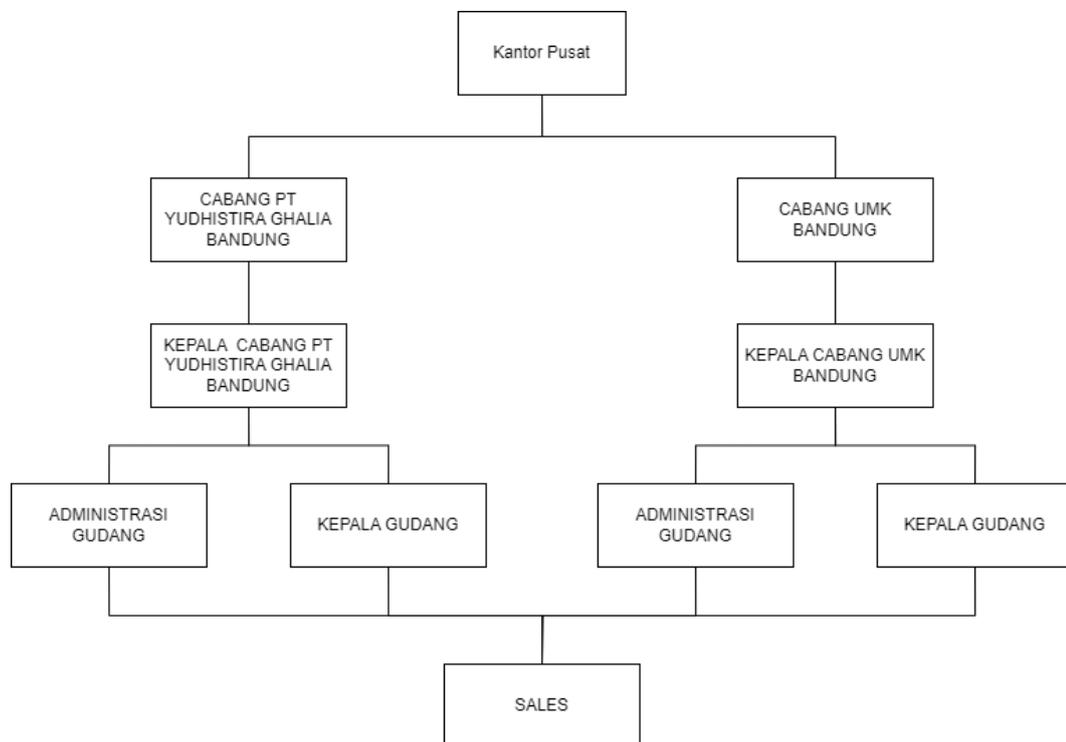
“Mendarmabaktikan diri pada dunia perbukuan untuk berperan serta mencerdaskan kehidupan bangsa guna meningkatkan kualitas sumber daya manusia Indonesia.”

2.3.2 Misi

1. Menciptakan buku bermutu untuk memenuhi sarana Pendidikan bangsa
2. Memenuhi kebutuhan pelanggan dengan memberikan pelayanan terbaik melalui jaringan distribusi yang kuat dan luas
3. Menumbuhkan Perusahaan yang sehat secara berkesinambungan dari tahun ke tahun.

2.4 Struktur Organisasi

Struktur organisasi dalam suatu perusahaan merupakan hal yang sangat penting, dengan adanya struktur organisasi ini memberikan pembagian tugas sesuai dengan bidangnya masing-masing. Struktur organisasi pada PT Yudhistira Ghalia cabang UMK Bandung diberikan pada Gambar 2. 1 yaitu:



Gambar 2. 1 Struktur Organisasi PT Yudhistira Ghalia cabang UMK Bandung

Adapun tugas dan deskripsi kerja dari masing-masing jabatan yaitu:

1. Kepala Cabang PT Yudhistira Ghalia Bandung
Bertugas sebagai kepala cabang yang mendistribusikan Buku SD, SMP, dan SMA, lalu mengambil Keputusan bisnis, memvalidasi ketika ada dokumen

ataupun faktur yang akan dibuat, memvalidasi surat permintaan serta mengakomodir bagian administrasi dan kepala Gudang.

2. Kepala cabang UMK Bandung

Bertugas sebagai kepala cabang yang mendistribusikan Buku SMK, lalu mengambil Keputusan bisnis, memvalidasi ketika ada dokumen ataupun faktur yang akan dibuat, memvalidasi surat permintaan serta mengakomodir bagian administrasi dan kepala gudang.

3. Administrasi Gudang

Membuat surat permintaan, menginput data kedalam sistem informasi Perusahaan (MISS), berkoordinasi dengan kepala gudang ketika ada permintaan buku, melakukan segala pencatatan buku yang keluar dan masuk.

4. Kepala Gudang

Bertugas ketika ada pesanan maka buku akan langsung disiapkan oleh kepala gudang, melakukan stock opname 3 bulan sekali, menerima barang ketika ada kiriman buku dari kantor pusat.

5. *Sales*

Bertugas untuk penjualan dan promosi ke sekolah sekolah yang ada di bandung raya, melakukan kerjasama dengan sekolah untuk pembelian buku.

Dari tugas dan deskripsi diatas, maka yang akan terlibat dalam sistem yang berjalan adalah jabatan administrasi gudang. Administrasi gudang fokus pada penggunaan sistem informasi perusahaan, sehingga sistem yang akan dibuat bisa digunakan oleh administrasi gudang.

2.5 Landasan Teori

Landasan teori pada penelitian kali ini membahas mengenai materi atau teori apa saja yang digunakan sebagai acuan dalam membuat aplikasi Strategi Penjualan PT Yudhistira Ghalia Indonesia cabang UMK Bandung. Landasan teori diuraikan meliputi teori umum, teori analisis sistem, teori perancangan sistem, dan teori pengujian sistem.

Teori umum membahas konsep strategi penjualan, dan teknik *bundling*. Teori analisis sistem membahas mengenai teknik *data mining*, metode *association rule*,

dan algoritma *FP-Growth*. Teori perancangan sistem berisi teori tentang *Unified Modelling language (UML)*, *use case*, *activity diagram*, *class diagram*, *sequence diagram*, dan bahasa pemrograman *python*. Teori pengujian sistem berisi teori tentang *black box testing*, nilai *lift ratio*, dan *user acceptance test*. Landasan teori didapat dari hasil studi literatur dengan mengumpulkan data dari jurnal, *browsing*, referensi buku, dan bacaan lain yang berhubungan dengan penelitian.

2.5.1 Strategi Penjualan

Strategi penjualan adalah rencana yang menyatukan, menyeluruh, dan terpadu dan melibatkan keunggulan strategi perusahaan dalam tantangan lingkungan yang dirancang untuk memastikan tujuan utama dari perusahaan dapat dicapai dengan pelaksanaan yang tepat. Selain itu, strategi penjualan adalah rencana yang dibuat oleh perusahaan untuk menentukan bagaimana mereka dapat meningkatkan volume penjualan produknya dan memenuhi dan memenuhi permintaan konsumen [10]. Setelah mengeksplorasi berbagai strategi penjualan yang dapat meningkatkan performa bisnis, pendekatan yang seringkali efektif dalam menarik minat konsumen dan meningkatkan penjual salah satunya yaitu strategi *bundling*. Strategi ini melibatkan penawaran beberapa produk atau layanan sebagai satu paket, yang seringkali menawarkan nilai yang lebih baik bagi pelanggan dan keuntungan yang lebih besar bagi penjual

2.5.2 Bundling

Gultinan mengatakan *bundling* adalah teknik pemasaran yang menggabungkan dua atau lebih produk dengan harga khusus [11]. *Bundling* juga bisa disebut sebagai cara menjual satu atau lebih barang dan jasa dalam satu paket dengan harga dan kemasan yang lebih menarik daripada menjualnya secara terpisah, yang membuatnya unik. perusahaan melakukan *bundling* produk untuk berbagai alasan, termasuk menghemat uang dan mendapatkan peluang pasar yang lebih besar. Dalam skala ekonomi, *bundling* produk dapat membantu mengurangi biaya pengemasan dan persediaan dibandingkan dengan memasarkan produk secara terpisah [12]. Menurut sebuah studi tahun 1997 oleh *Mercer Management Consulting, Lexington, Massachusetts*, *bundle* yang baik memiliki 5 kualitas yaitu: [13]

1. Paket tersebut memiliki nilai lebih daripada jumlah tiap-tiap unit
2. Bundel memberikan kesederhanaan dan kemudahan dalam memilih di antara berbagai pilihan
3. Bundel memecahkan masalah bagi konsumen
4. Bundel fokus dan bersandar dalam upaya untuk menghindari membawa pilihan konsumen yang tidak digunakan.
5. Bundel menarik perhatian atau bahkan kontroversi.

Pertimbangan yang digunakan untuk melakukan bundle dapat dilihat dari nilai atau *value*. *Value* dari *bundling* bisa dievaluasi dengan adanya teori *value creation* dapat dilihat dari empat dimensi yaitu: [13].

1. *Novelty*

Perusahaan memberikan produk pada konsumen dengan melakukan berbagai inovasi. Baik berupa paket penjualan ataupun strategi yang menarik konsumen.

2. *Lock-in*

Membuat pelanggan melakukan pembelian kembali dan bagaimana Perusahaan dapat mempertahankannya. Hal ini berfungsi agar pelanggan menjadi pelanggan setia.

3. *Complementary*

Memberikan nilai besar dari total nilai yang diberikan produk tersebut ketika dijual terpisah. Bisa berupa diskon maupun bonus ketika konsumen tersebut membeli dengan volume tertentu.

4. *Efficiency*

Dapat dicapai bila terdapat pengurangan biaya, baik transaksi, biaya pencarian, kecepatan dan kemudahan. Sehingga Perusahaan bisa menghemat biaya mereka.

Setelah memahami strategi *bundling* dan efektivitasnya dalam meningkatkan penjualan, selanjutnya perlu melihat bagaimana teknologi bisa mendukung strategi ini. Salah satu cara adalah dengan menerapkan data mining untuk menganalisis pola pembelian pelanggan. Dengan memanfaatkan *insight* dari *data mining*, maka dapat

dirancang paket *bundling* yang lebih menarik dan relevan sesuai dengan kebutuhan konsumen.

2.5.3 Data Mining

Data adalah kumpulan informasi atau fakta yang dibuat dengan kata, kalimat, simbol, angka, dan lainnya. Data dapat diperoleh melalui proses pencarian dan pengamatan yang tepat dari berbagai sumber. Data juga dapat didefinisikan sebagai kumpulan informasi atau deskripsi dasar yang berasal dari objek atau peristiwa tertentu. [14].

Data *mining* adalah proses menemukan hubungan baru yang memiliki arti, pola, dan kebiasaan dengan memilah-milah sebagian besar data yang disimpan dalam media penyimpanan dengan menggunakan teknologi pengenalan pola seperti matematika dan statistik. Data *mining* adalah gabungan dari beberapa disiplin ilmu yang menyatukan teknik pembelajaran mesin, pengenalan pola, statistik, *database*, dan *visualisasi* untuk menangani masalah pengambilan informasi dari *database* [15]. Dalam data mining ada beberapa metode yang sering digunakan yaitu klasifikasi, *clustering*, regresi, dan asosiasi. Pada metode asosiasi lebih dikenal dengan nama *association rule*. Berdasarkan tugas yang dapat dilakukannya, data *mining* dibagi menjadi beberapa kelompok, yaitu:

1. Deskripsi

Deskripsi menggambarkan pola, kecenderungan yang ada pada data, dan memungkinkan memberi penjelasan terhadap pola yang ada. Tujuannya memberikan penjelasan dari pola yang terdapat dalam data.

2. Klasifikasi

Klasifikasi adalah proses menemukan model dan fungsi yang membedakan kelas data dengan tujuan memprediksi kelas untuk data yang tidak diketahui kelasnya. Klasifikasi memiliki aplikasi yang luas termasuk deteksi penipuan, target pemasaran dan diagnosis medis.

3. Estimasi

Model ini dibangun menggunakan *record* lengkap yang menyediakan nilai variabel target sebagai nilai prediksi. Estimasi digunakan untuk memprediksi nilai dari suatu variabel berdasarkan fitur yang ada.

4. Prediksi

Prediksi dalam data mining adalah proses untuk memprediksi nilai dari fitur yang diberikan. Prediksi dapat digunakan untuk membuat suatu perencanaan dari suatu kejadian.

5. Pengklusteran

Merupakan pengelompokkan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek yang memiliki kemiripan. Pengklusteran mencoba untuk membentuk kelompok berdasarkan pola kemiripan dari setiap data atau sampel.

6. *Association rule*

Association rule dalam data mining adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja. *Association rule* seringkali digunakan ketika menghadapi masalah bisnis, misalnya analisis transaksi penjualan dan menganalisis pola pembelian dari konsumen.

Dalam penelitian ini akan digunakan pendekatan *association rule* dari *data mining*. *Association rule* akan digunakan untuk penentuan pola informasi dan penentuan rule yang terbentuk.

2.5.4 Association Rule

Association rule adalah salah satu metode data mining yang digunakan untuk menemukan hubungan kesamaan antar item. Dalam metode *association rule*, ada istilah *itemset*, yang merujuk pada sekumpulan item. Untuk menentukan apakah sebuah *itemset* dapat menjadi *association rule*, *itemset* tersebut harus memenuhi ambang batas *minimum support* dan *minimum confidence* [16]. Proses pencarian *association rule* terbagi menjadi dua tahap, yaitu analisis *frequent itemset* dan pembentukan *association rule* [7]. Pencarian *association rule* dilakukan melalui:

1. Analisis *Frequent itemset*

Tahapan ini mencari kombinasi item yang memenuhi syarat *minimum* dari nilai *support* dalam *database*. Nilai *support* sebuah item diperoleh dengan rumus (2.1) sebagai berikut: [7]

$$Support(x) = \frac{\text{jumlah transaksi yang mengandung item } x}{\text{total transaksi}} \times 100\% \quad (2.1)$$

Jika akan dicari nilai dari dua item, maka digunakan rumus (2.2) berikut:

$$Support(x, y) = \frac{\sum \text{Transaksi yang mengandung item } x \text{ dan } y}{\sum \text{total transaksi}} \times 100 \quad (2.2)$$

Keterangan :

x : Item Antecedent (sebab)

y : Item Consequent (akibat)

Persamaan (2.2) dapat diperluas untuk menghitung nilai *support* untuk kombinasi tiga item atau lebih. Setelah semua *frequent itemset* didapat, ditentukan *syarat minimum confidence*. *Confidence* adalah ukuran yang menunjukkan seberapa valid suatu association rule. *Confidence* suatu rule adalah probabilitas bahwa beberapa produk dibeli secara bersamaan, dengan asumsi bahwa salah satu produk tersebut sudah pasti dibeli dengan mempertimbangkan nilai ambang batas *minimum confidence*.

2. Pembentukan *association rule*

Setelah semua pola *frequent itemset* ditemukan, kemudian dicari aturan *association rule* yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiatif ($x \rightarrow y$). nilai *confidence* dari aturan $x \rightarrow y$ diperoleh dari rumus (2.3) berikut: [7]

$$Confidence(x \rightarrow y) = \frac{\sum \text{support count } (x \cup y)}{\sum \text{support count } x} \quad (2.3)$$

Biasanya, aturan asosiasi dianggap menarik jika memenuhi *minimum confidence* dan *minimum support*. Nilai *minimum* ini dapat ditetapkan oleh pengguna atau ahli domain. Analisis tambahan dapat dilakukan untuk menemukan statistic yang menarik untuk menemukan korelasi antara item-item yang terkait [7]

Metode *association rule* membutuhkan suatu algoritma untuk mencari pola informasi dan rule yang akan terbentuk. Salah satu algoritma yang dipakai adalah *algoritma FP-Growth*.

2.5.5 Algoritma *FP-Growth*

FP-Growth adalah algoritma yang dapat digunakan untuk menentukan himpunan data yang sering muncul (*frequent itemset*) dalam kumpulan data tanpa menggunakan *candidate generation*. *FP-Growth* adalah *evolusi* dari algoritma *apriori*, perbedaannya terletak pada kecepatan *scanning database* dan akurasi peraturannya [17]. Salah satu langkah dari algoritma *Fp-Growth* yaitu *Fp-Tree*. *FP-Tree* adalah struktur penyimpanan data yang dimampatkan dan dibangun dengan memetakan setiap data transaksi ke dalam setiap jalur *FP-tree* tertentu. Jalur *FP-tree* memungkinkan saling menimpa karena ada transaksi dengan item yang sama di setiap lintasan *FP-tree* yang dipetakan. Dengan demikian, semakin banyak data transaksi dengan item yang sama yang dipetakan, semakin baik proses pemampatan *FP-tree* [18]. Langkah-langkah dalam algoritma *FP-Growth* terdiri dari beberapa tahap yaitu :

1. Membuat *header* item

Header sebagai jenis item dasar yang memenuhi *minimum support*. Setelah mendapatkan item dan nilai support-nya, maka item yang tidak *Frequent* dibuang dan item yang lain diurutkan berdasarkan nilai *support*-nya. *Header* untuk item, disiapkan untuk suatu *array* dan ditambahkan saat membentuk *FP-Tree*.

2. Membuat *FP-Tree*

FP-Tree dibuat dengan mencari item sesuai urutan pada item *frequent* lalu ketika item sudah diurutkan bisa langsung dimasukkan ke dalam stuktur *FP-Tree*. Setelah itu membuat *root*, dan tiap item yang ditemukan dimasukkan berdasarkan *path* nya. Jika item yang ditemukan sudah ada, maka nilai *support* item tersebut yang ditambahkan. Namun saat *path* belum terdapat nilainya, maka dibuat *node* baru untuk melengkapi *path* baru tersebut. Hal itu hanya dilakukan ketika item pada transaksi memenuhi nilai *minimum support*.

3. *Pattern extraction*

Pattern extraction dilakukan berdasarkan keterlibatan item pada suatu *path*. Pada setiap *path*, akan diperiksa semua kombinasi item yang mungkin

terlibat, lalu dilanjutkan ke iterasi berikutnya dan melibatkan item selanjutnya. Jika item pertama suatu kombinasi bukan item terakhir, maka kombinasi *itemset* tersebut masih bisa dilanjutkan ke iterasi yang berikutnya.

4. Mengurutkan dan menyeleksi *pattern*

Pattern yang tidak memenuhi *minimum support*, dihapus dari daftar *pattern*. *Pattern* yang tersisa kemudian diurutkan untuk memudahkan pembuatan *rules*.

2.5.6 *Unified Modelling Language (UML)*

Bahasa pemodelan perangkat lunak yang distandarisasi sebagai media penulisan cetak biru (*Blueprints*) adalah *Unified Modeling Language (UML)*. UML dapat digunakan untuk mendokumentasikan visualisasi, spesifikasi, konstruksi, dan dokumentasi beberapa komponen sistem yang ada dalam perangkat lunak. [19]. Dalam penelitian ini UML dapat digunakan untuk mendukung perancangan sistem. UML mempunyai sehingga berbagai jenis diagram untuk mendukung dalam perancangan sistem, yaitu *use case*, *sequence diagram*, *activity diagram*, *class diagram* yang membantu penulis dalam membangun perangkat lunak. Penjelasan diagram yang ada pada UML diantaranya yaitu:

1. *Use case*

Sebuah *use case* menjelaskan bagaimana sebuah sistem berfungsi dari sudut pandang pengguna [20]. *Use case* mendefinisikan hal apa saja yang akan diproses oleh sistem, bekerja dengan *scenario*, yang mencakup langkah-langkah apa saja yang dilakukan oleh user terhadap sistem atau sebaliknya. *Use case* juga mengidentifikasi fitur yang ada pada sistem, cara *user* berinteraksi dengannya, dan bagaimana fitur berinteraksi dengan fungsi sistem.

2. *Activity Diagram*

activity diagram menggambarkan ide-ide tentang aliran data, kontrol, dan aksi yang dirancang dengan baik dalam suatu sistem [21]. Setelah membuat model *use case*, maka setiap *scenario* yang ada di *use case* akan dideskripsikan lebih jelas di dalam *activity diagram*. Sebuah *activity*

diagram merupakan pemodelan yang menggambarkan sistem kerja sebuah objek atau sistem. Dalam *activity diagram*, alur proses kerja terstruktur dari titik awal hingga titik akhir digambarkan, dengan setiap aktivitas diberi notasi sesuai fungsinya.

3. *Class Diagram*

Salah satu pemodelan UML yang paling penting adalah *class diagram*, yang digunakan untuk membuat *logical model* dari sebuah sistem. *Class diagram* menunjukkan bagaimana skema arsitektur sistem yang sedang dirancang. Dalam *class diagram*, sebuah *class* memiliki atribut dan metode. setiap *class* dihubungkan dengan sebuah garis yang disebut *association rule* [22].

4. *Sequence diagram*

Dalam perancangan antarmuka pemakai, *sequence diagram* dapat digunakan untuk dua tujuan. Pertama, mereka dapat menguraikan sebuah proses bisnis menjadi aktivitas-aktivitas yang lebih kecil, dan kedua, mereka dapat menentukan tingkat interaksi pemakai yang diperlukan untuk masing-masing aktivitas [23]. Keunggulan dari *sequence diagram* adalah cara sederhana dan efektif dalam mengkomunikasikan urutan peristiwa dalam sebuah interaksi.

2.5.7 *Python*

Bahasa pemrograman yang dapat mendukung dalam perancangan dan pembuatan sistem salah satunya adalah Bahasa pemrograman *python*. *Python* adalah bahasa pemrograman interpretatif yang dapat digunakan untuk berbagai tujuan dan filosofi perancangannya berpusat pada tingkat keterbacaan kode. Python digambarkan sebagai bahasa yang memiliki banyak kemampuan bersama dengan sintaksis kode yang sangat jelas. Selain itu, itu memiliki banyak fitur pustaka standar yang luas dan komprehensif. *Python* dapat digunakan sebagai bahasa pemrograman dinamis dengan manajemen memori otomatis, seperti halnya bahasa pemrograman dinamis lainnya. Namun, *python* biasanya digunakan sebagai bahasa *script* untuk mencapai tujuan yang biasanya tidak dapat dicapai dengan bahasa *script* [24]. *Library* yang digunakan pada sistem yang akan dibangun adalah *mlxtend*. Kode program menghitung itemset menggunakan algoritma *FP-Growth*

dari library mlxtend, mlxtend mengambil dataset yang berisi daftar item, lalu mengubahnya menjadi menggunakan *class transaction encoder* dari mlxtend, kemudian menerapkan fungsi *FP-Growth* untuk mengekstrak itemset yang sering muncul. Selain memudahkan untuk kebutuhan proses data, *python* dapat digunakan untuk hal lain seperti membuat *website* dan tampilan GUI (*Graphical User Interface*).

GUI yang seringkali digunakan adalah PyQt, PyQt adalah lintas platform GUI perangkat qt yang berhubungan pada python yang di implementasikan sebagai plug-in. PyQt tersedia dibawah persyaratan yang sama untuk Qt versi yang lebih tua dari 4,5. Versi PyQt yang digunakan dalam penelitian kali ini adalah PyQt6, karena versi tersebut lebih baru dengan fitur yang lebih banyak, PyQt6 mempunyai dukungan terhadap *widget* dan fitur, sehingga menjadikan PyQt6 salah satu pustaka *python* yang populer dalam membuat GUI.

2.5.8 Lift Ratio

Lift ratio adalah nilai perbandingan antara nilai *confidence* dengan *benchmark confidence*. *Benchmark confidence* sendiri adalah perbandingan antara jumlah semua item *consequent* terhadap total jumlah transaksi. *Lift ratio* digunakan dalam mengukur tingkat penting atau tidaknya suatu *rule* yang terbentuk berdasarkan nilai *support* dan *confidence* [25]. Rumus untuk mencari nilai *benchmark confidence* dan *lift ratio* masing masing dapat dilihat pada rumus ((2.4) dan ((2.5) berikut ini:

$$\text{Benchmark Confidence} = \frac{Nx}{N} \quad (2.4)$$

$$\text{Lift Ratio} = \frac{\text{Confidence}(x,y)}{\text{Benchmark Confidence}(x,y)} \quad (2.5)$$

Keterangan :

Nx : Jumlah transaksi dengan item yang menjadi akibat (*consequent*)

N : Total transaksi pada database

x, y : Item antecedent dan consequent

lift ratio dapat menentukan aturan yang terbentuk valid atau tidak. Nilai *lift ratio* dimulai dari nol hingga seterusnya, jika nilai *lift ratio* lebih besar daripada 1 maka aturan tersebut dinyatakan valid dan menandakan kekuatan asosiasinya besar, sedangkan jika nilai *lift ratio* kurang dari 1 maka aturan tersebut kekuatan asosiasinya kecil.

2.5.9 *Black Box testing*

Pengujian fungsional program perangkat lunak atau sistem yang diuji dikenal sebagai "*black box*" pengujian ini dilakukan berdasarkan spesifikasi yang diperlukan dan tidak memerlukan pemeriksaan kode, ini dilakukan dari sudut pandang pengguna dan hanya melakukan pengujian yang mengetahui kumpulan masukan dan keluaran yang dapat diprediksi. Metode *black box testing* berfokus pada spesifikasi fungsional perangkat lunak, penguji dapat mengidentifikasi kumpulan kondisi *input* dan menguji spesifikasi fungsional program [26]. Pengujian *black box* memiliki keuntungan besar karena membantu mengungkap ambiguitas dan ketidakkonsistenan dalam spesifikasi persyaratan. Berikut adalah Langkah-langkah dalam pengujian black box menurut Doroty Graham, Erik van veenendaal, Isabel evans, dan rex black, yaitu [27]:

1. Identifikasi persyaratan dan spesifikasi

Kumpulkan dan pahami dokumentasi persyaratan dan spesifikasi perangkat lunak yang akan diuji, yaitu spesifikasi kebutuhan fungsional dan non fungsional.

2. Identifikasi fungsi yang akan diuji

Daftarkan semua fungsi utama dan sub-fungsi perangkat lunak. Prioritaskan fungsi berdasarkan resiko dan pentingnya bagi pengguna

3. Desain kasus uji

Decision table testing menggunakan tabel keputusan untuk menangkap dan menguji kombinasi dari kondisi inputan

4. Menyiapkan lingkungan pengujian

Siapkan lingkungan yang diperlukan untuk pengujian, termasuk perangkat keras, perangkat lunak, jaringan dan alat pengujian.

5. Menjalankan kasus uji

Eksekusi kasus uji yang telah dirancang, kemudian mencatat hasil pengujian termasuk hasil aktual dan hasil yang ingin diharapkan.

6. Dokumentasi dan pelaporan

Dokumentasikan proses pengujian, hasil, dan Kesimpulan. Buat laporan akhir pengujian yang mencakup ringkasan pengujian, defect yang ditemukan, dan rekomendasi.

Berikut adalah contoh dari dokumen *black box testing*[28]:

Tabel 2. 1 Dokumen black box testing

Test Case	Deskripsi test case	Test step	Excepted results	Actual results	Pass / Fail

2.5.10 User Acceptance Test

Menurut Perry, William E, User Acceptance Testing (UAT) merupakan Pengujian Pengakuan Pengguna (UAT) dilakukan oleh end-user, yang merupakan karyawan atau staf perusahaan yang langsung berinteraksi dengan sistem. UAT melakukan verifikasi apakah fungsi yang ada berjalan sesuai dengan kebutuhan atau fungsinya. Setelah proses pengujian selesai, hasil pengujian penerimaan menunjukkan bahwa sistem perangkat lunak memenuhi persyaratan [29]. User acceptance test merupakan suatu cara untuk mengetahui seberapa mudah aplikasi digunakan oleh pengguna [30]. User acceptance test mempunyai tujuan yaitu memastikan bahwa aplikasi yang sedang dibangun dan dikembangkan dapat memenuhi persyaratan dan kebutuhan pengguna atau memenuhi tujuan bisnis dari pengguna. User acceptance test adalah fase terakhir dari proses pengujian perangkat lunak. Selama UAT, perangkat lunak diuji untuk memastikan bahwa tugas-tugas sudah memenuhi spesifikasinya. UAT adalah salah satu prosedur proyek perangkat lunak paling penting yang harus dilakukan sebelum perangkat lunak dikembangkan dan dirilis ke pasar [31]. Pengujian sistem bertujuan untuk menguji sistem dari sudut pandang fungsional sistem, yang menentukan apakah sistem beroperasi dengan benar dan apakah hasil yang diberikan memenuhi harapan. Pengujian user

acceptance test menggunakan teknik black box untuk menguji sistem terhadap spesifikasinya