

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan revolusi industri 4.0 yang berlangsung saat ini, banyak aspek kehidupan masyarakat mengalami perubahan signifikan. Berbagai aktivitas bisnis yang sebelumnya dilakukan secara manual kini berinovasi dengan memanfaatkan teknologi, agar usaha yang dijalankan dapat mencapai tujuan secara maksimal, bekerja lebih efisien, serta mampu menghadapi persaingan usaha yang semakin ketat (Tarantang et al., 2024).

Klinik Pratama Tanjung merupakan salah satu klinik kesehatan ibu dan anak yang menawarkan berbagai jenis layanan, seperti persalinan, baby spa, dan USG kehamilan. Beragamnya layanan, seiring dengan meningkatnya jumlah pasien, membuat volume data yang harus dikelola oleh bagian administrasi klinik menjadi sangat besar, terutama terkait administrasi layanan kesehatan. Tidak jarang terjadi kesalahan atau error dalam pencatatan transaksi layanan kesehatan, yang menimbulkan kerugian baik dari sisi keuangan maupun administrasi klinik.

Berdasarkan wawancara yang dilakukan pada 7 Mei 2025 dengan Febri selaku pengelola Klinik Pratama Tanjung, diketahui bahwa sistem pelayanan pasien dan pengelolaan data transaksi di klinik ini belum berjalan cukup efektif, karena masih sering terjadi kesalahan pencatatan. Kesalahan tersebut kerap terungkap saat manajemen klinik melakukan pemeriksaan data kunjungan rutin pada akhir minggu.

Melihat permasalahan tersebut, Klinik Pratama Tanjung membutuhkan solusi berupa sarana yang dapat membantu petugas administrasi dalam mencatat transaksi layanan kesehatan, layanan yang diberikan kepada pasien, serta pengelolaan penerimaan pembayaran setelah layanan selesai. Solusi yang diusulkan adalah sistem informasi berbasis aplikasi mobile, yang mampu mencatat seluruh transaksi setiap layanan secara terintegrasi. Sistem informasi sendiri merupakan kumpulan subsistem yang saling terhubung dan berkolaborasi untuk memecahkan masalah tertentu dengan cara mengolah data agar menghasilkan nilai tambah yang bermanfaat bagi penggunanya. Pemanfaatan sistem informasi berbasis mobile diharapkan dapat meningkatkan kualitas pelayanan, manajemen, dan promosi klinik. Solusi berbasis aplikasi mobile ini juga didukung oleh fakta bahwa mayoritas pasien Klinik Pratama Tanjung merupakan ibu-ibu muda yang umumnya telah memiliki dan terbiasa menggunakan smartphone.

Solusi yang dibutuhkan Klinik Pratama Tanjung tersebut diwujudkan melalui inovasi berupa pengembangan aplikasi transaksi layanan kesehatan berbasis mobile. Aplikasi ini dirancang untuk memudahkan pencatatan data transaksi pada layanan kesehatan secara terintegrasi dengan sistem klinik.

Namun demikian, hingga saat ini belum dapat dipastikan apakah penerapan aplikasi layanan kesehatan berbasis mobile tersebut benar-benar mampu mengurangi kemungkinan terjadinya kesalahan pencatatan layanan kesehatan. Pada dasarnya, penerapan suatu sistem di dalam sebuah usaha selalu menghadapi dua kemungkinan, yakni keberhasilan atau kegagalan sistem (DeLone, 1988). Oleh

karena itu, diperlukan penelitian lebih lanjut untuk membuktikan efektivitas sistem yang dikembangkan.

1.2 Perumusan masalah

1. Bagaimana aplikasi Transaksi layanan kesehatan dapat membantu layanan kesehatan di Klinik Pratama Tanjung menjadi lebih moderen
2. Bagaimana aplikasi transaksi layanan kesehatan dapat membantu pengolahan data klinik menjadi lebih rapih dan efisien.
3. Bagaimana merancang dan membangun aplikasi transaksi layanan kesehatan berbasis mobile pada Klinik Pratama Tanjung?
4. Bagaimana merancang sistem transaksi layanan kesehatan menggunakan algoritma greedy

1.3 Batasan Masalah

Beberapa Batasan masalah yang terdapat dalam skripsi ini:

1. Data yang dikelola hanya sebatas data administrasi layanan kesehatan, tidak mencakup rekam medis mendetail pasien.
2. Fitur aplikasi meliputi data obat, pencatatan transaksi layanan kesehatan, manajemen data obat.
3. Rancangan bangun sistem transaksi layanan kesehatan ini menggunakan bahasa java dengan software eclipse versi juno.
4. Rancangan sistem transaksi layanan kesehatan ini menggunakan algoritma greedy dalam sistem kembalian.

1.4 Tujuan dan Manfaat Penelitian

1.4.1 Tujuan Penelitian

Berdasarkan perumusan masalah, maka tujuan penelitian yang akan di capai sebagai berikut:

1. Merancang dan membangun aplikasi transaksi layanan kesehatan berbasis mobile untuk Klinik Pratama Tanjung.
2. Menghasilkan aplikasi yang mampu mempermudah pencatatan transaksi layanan kesehatan secara cepat, tepat, dan terintegrasi.
3. Mengurangi risiko kesalahan pencatatan transaksi dan meningkatkan efisiensi kerja di bagian administrasi klinik.
4. Menghasilkan pecahan kembalian uang dengan menggunakan algoritma greedy

1.4.2 Manfaat Penelitian

1. Menciptakan sistem proses transaksi yang dapat di nikmati juga oleh pasien
2. Membantu mengurangi penumpukan berkas yang di akibatkan masih digunakannya penginputan manual.
3. Membantu Klinik Pratama Tanjung menjadi lebih efisien dalam penginputan data.

1.5 Metodologi Penelitian

1. Pengumpulan Data Obat

Pengumpulan data obat dilakukan untuk menunjang pengembangan fitur manajemen obat pada aplikasi. Data dikumpulkan melalui wawancara dengan petugas klinik, observasi terhadap proses pencatatan manual, dan dokumentasi dari daftar obat yang tersedia. Informasi yang dikumpulkan meliputi nama obat, jenis, harga, dan jumlah stok. Data ini kemudian digunakan sebagai dasar perancangan database serta sistem pencatatan dan pengelolaan obat dalam aplikasi mobile yang dikembangkan.

2. Perancangan

Perancangan proses transaksi berbasis *mobile* menggunakan Java android.

3. Pengkodean

Pada tahap ini dilakukan pengkodean aplikasi transaksi layanan kesehatan dalam proses transaksi dan pendataan pada Klinik Pratama Tanjung dengan perancangan yang telah ditentukan.

4. Pengujian Aplikasi

Pada tahap ini dilakukan pengujian terhadap aplikasi *Mobile* dalam proses transaksi agar tidak terjadinya bug ataupun kesalahan sistem.

5. Penyusunan Laporan

Pada tahap ini dilakukan penulisan dokumen dan laporan dari aplikasi Transaksi Layanan Kesehatan pada klinik pratama tanjung dalam proses transaksi di Klinik Pratama Tanjung.

1.6 Sistem penulisan

Sistem penulisan tugas skripsi ini dibagi ke dalam beberapa bab, yang masing-masing bab terdiri atas beberapa subbab untuk mempermudah penjelasan mengenai penelitian yang dilakukan serta memudahkan pembaca dalam memahami isi penelitian. Adapun sistematika penulisan tugas skripsi ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan, manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan skripsi.

BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori dan pengetahuan dasar yang diperoleh dari pengumpulan data serta dokumentasi literatur, termasuk sumber-sumber dari internet, yang digunakan untuk memahami permasalahan yang dibahas dalam penelitian ini.

BAB III METODE PENELITIAN

Bab ini menguraikan tahapan-tahapan sistematis yang digunakan dalam pelaksanaan penelitian. Tahapan-tahapan tersebut menjadi kerangka yang dijadikan pedoman penelitian untuk mencapai tujuan yang telah ditetapkan. Tahapan tersebut mencakup waktu dan tempat penelitian, serta alat dan bahan yang digunakan dalam pengembangan aplikasi transaksi layanan kesehatan pada Klinik Pratama Tanjung.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi hasil dan pembahasan mengenai aplikasi mobile transaksi layanan kesehatan pada Klinik Pratama Tanjung dalam konteks sistem transaksi di klinik tersebut.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari keseluruhan uraian penulisan skripsi, serta saran yang diajukan untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi, secara umum, adalah subkelas dari perangkat lunak komputer yang memanfaatkan kemampuan komputer secara langsung untuk melakukan tugas yang diinginkan oleh pengguna. Aplikasi juga dapat diartikan sebagai penerjemah perintah-perintah yang dijalankan oleh pengguna komputer untuk diteruskan atau diproses oleh perangkat keras. Selain membantu dan mempercepat proses pekerjaan manusia, aplikasi juga dapat menghasilkan output yang lebih akurat dalam memecahkan masalah.

Pada umumnya, aplikasi digunakan untuk mengontrol perangkat keras (device driver), melakukan proses perhitungan, serta berinteraksi dengan perangkat lunak lainnya, seperti sistem operasi dan bahasa pemrograman (Wahyuni et al., 2022).

Pengertian Aplikasi Menurut Para Ahli:

- a. Menurut Sri Widiанти, aplikasi adalah perangkat lunak yang dibuat sebagai front-end sebuah sistem yang digunakan untuk mengelola data sehingga menjadi informasi yang berguna bagi pengguna.
- b. Menurut Rachmad Hamim, aplikasi adalah perangkat lunak (software) yang dibuat untuk tujuan tertentu, seperti mengolah dokumen, hiburan, dan lain sebagainya.
- c. Menurut Hengky Pratama, aplikasi adalah perangkat lunak yang dibuat khusus untuk memenuhi kebutuhan berbagai aktivitas dan pekerjaan.

2.1.1 Klasifikasi Aplikasi

1. Ada beberapa klasifikasi dalam aplikasi, Antara lain:

System software

2. Aplikasi yang dapat mengelola dan mengendalikan proses operasi internal di dalam sistem computer.

3. *Real Time Software*

Aplikasi yang memiliki fungsi untuk mengamati, menganalisa, serta mengendalikan suatu kejadian di dunia nyata saat sedang berlangsung.

4. *Business Software*

Aplikasi yang memiliki fungsi untuk mengamati, menganalisa, serta mengendalikan suatu kejadian di dunia nyata saat sedang berlangsung.

5. *Engineering and Scientific Software*

Aplikasi yang diciptakan dan dikembangkan untuk membantu manusia menyelesaikan suatu masalah yang sifat nya non algoritmik.

6. *Web Based Software*

Aplikasi yang memiliki fungsi sebagai media penghubung antara pengguna dengan internet secara langsung.

7. *Personal Computer Software*

Aplikasi yang digunakan untuk perangkat pengguna resmi dan juga pribadi yang sudah banyak digunakan saat ini.

2.1.2 Fungsi Aplikasi

Fungsi aplikasi sebagai berikut:

1. Perangkat lunak perusahaan (*enterprise*)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media dan hiburan
5. Perangkat lunak pengembangan media
6. Perangkat lunak pendidikan
7. Perangkat lunak rekayasa produk

2.2 Aplikasi *Mobile*

2.2.1 Pengertian Aplikasi *Mobile*

Aplikasi *mobile* adalah perangkat lunak yang dirancang untuk dijalankan pada perangkat seluler, seperti smartphone atau tablet. Aplikasi ini dibuat khusus untuk memanfaatkan fitur perangkat seluler, seperti layar sentuh, kamera, GPS, sensor, dan koneksi internet. Aplikasi *mobile* dapat berfungsi untuk berbagai keperluan, termasuk komunikasi, hiburan, produktivitas, pendidikan, kesehatan, dan transaksi bisnis. (Ikhsan et al., 2020)

Ada beberapa pengertian website menurut para ahli:

1. Menurut Brian Fling, dalam bukunya “*Mobile Design and Development*” aplikasi *mobile* adalah perangkat lunak yang dirancang untuk berjalan di perangkat portabel seperti smartphone dan tablet, dan dibangun untuk

memanfaatkan fitur spesifik yang ditawarkan oleh perangkat tersebut seperti GPS, sensor, kamera, dan layar sentuh.

2. David R. Brown, Dalam penelitiannya di jurnal “*Mobile Technology in Education*” Aplikasi *mobile* adalah program atau perangkat lunak yang dijalankan di perangkat *mobile*, yang berfungsi untuk membantu pengguna menyelesaikan tugas tertentu dengan cepat dan efisien melalui akses cepat ke alat, informasi, atau layanan.
3. Jason Hong dan James A. Landay, Dalam penelitian mereka tentang interaksi manusia-komputer di lingkungan *mobile*, Aplikasi *mobile* adalah perangkat lunak yang memanfaatkan kemampuan *mobile* computing yang memungkinkan pengguna berinteraksi dengan perangkat komputasi kecil, portabel, dan nirkabel, seringkali dalam konteks mobilitas pengguna.

Secara umum, para ahli sepakat bahwa aplikasi *mobile* adalah perangkat lunak yang dirancang khusus untuk perangkat portabel seperti smartphone dan tablet, yang memanfaatkan fitur dan kemampuan perangkat *mobile* untuk menyediakan berbagai layanan atau fungsi, termasuk komunikasi, hiburan, produktivitas, dan komersial. Perbedaan utama antara aplikasi *mobile* dengan aplikasi desktop adalah aspek mobilitas, aksesibilitas, dan optimalisasi pengalaman pengguna yang diadaptasi dengan lingkungan *mobile*.

2.2.2 Jenis-Jenis Aplikasi *Mobile*

Aplikasi *mobile* dapat dibedakan menjadi beberapa jenis berdasarkan teknologi pengembangannya dan tujuan penggunaannya. Berikut adalah jenis-jenis aplikasi *mobile*:(Mastan, 2021) Berdasarkan Sifat :

1. Aplikasi Native

Aplikasi yang dikembangkan untuk satu platform atau sistem operasi tertentu, seperti Android atau iOS. Aplikasi native dibuat menggunakan bahasa pemrograman yang didukung oleh platform tersebut, seperti:

- a. **Android:** Java atau Kotlin
- b. **iOS:** Swift atau Objective-C

Aplikasi native menawarkan kinerja yang optimal dan akses penuh ke fitur perangkat, seperti kamera, sensor, GPS, dan lainnya.

2. Aplikasi Web *Mobile*

Aplikasi ini dijalankan melalui browser di perangkat *mobile*. Tidak memerlukan instalasi atau pengunduhan dari toko aplikasi. Dibangun menggunakan teknologi web seperti HTML, CSS, dan JavaScript, aplikasi web ini dapat diakses di berbagai perangkat melalui URL.

Ciri utama aplikasi web adalah:

- a. Tidak tersedia offline (kecuali dengan dukungan tertentu seperti *service workers*).
- b. Desain responsif, sehingga menyesuaikan tampilan dengan layar perangkat.

3. Aplikasi Hybrid

Aplikasi hybrid menggabungkan elemen dari aplikasi web dan aplikasi native. Aplikasi ini dikembangkan menggunakan teknologi web tetapi dibungkus dalam aplikasi native. Beberapa kerangka kerja yang digunakan dalam pembuatan aplikasi hybrid adalah:

a. React Native

b. Ionic

c. Flutter

Aplikasi hybrid memungkinkan pengembang membuat aplikasi untuk beberapa platform dengan satu kode sumber, yang membuatnya lebih efisien dari segi waktu dan biaya pengembangan, meskipun terkadang tidak secepat aplikasi native. Berdasarkan tujuan:

1. Meningkatkan Kemudahan Akses dan Mobilitas

Salah satu tujuan utama aplikasi *mobile* adalah memberikan kemudahan akses informasi atau layanan kapan saja dan di mana saja. Karena perangkat *mobile* selalu tersedia dan mudah dibawa, pengguna dapat dengan cepat mengakses aplikasi untuk menyelesaikan tugas atau mendapatkan informasi tanpa harus bergantung pada komputer desktop.

2. Meningkatkan Efisiensi dan Produktivitas

Aplikasi *mobile* dirancang untuk meningkatkan efisiensi pengguna dalam melakukan tugas-tugas sehari-hari. Aplikasi ini menyediakan alat dan layanan yang mempermudah penyelesaian tugas, baik itu dalam hal pekerjaan, pendidikan, atau kebutuhan pribadi. Aplikasi ini juga membantu pengguna

untuk mengotomatisasi tugas-tugas manual dan mengelola waktu dengan lebih baik.

3. Meningkatkan interaksi dan komunikasi

Aplikasi *mobile* berfungsi sebagai alat untuk membangun interaksi dan komunikasi yang lebih cepat dan efektif antara individu, kelompok, atau organisasi. Fitur seperti pesan instan, panggilan video, dan media sosial memungkinkan pengguna untuk tetap terhubung secara real-time, meskipun mereka berada di tempat yang berbeda.

4. Memberikan Pengalaman Pengguna (*User Experience*) yang lebih baik

Tujuan lain dari aplikasi *mobile* adalah untuk memberikan pengalaman pengguna yang lebih baik dengan antarmuka yang disesuaikan dengan layar sentuh dan fitur perangkat seluler lainnya. Aplikasi *mobile* biasanya dirancang dengan antarmuka yang lebih intuitif dan navigasi yang lebih mudah dibandingkan dengan situs web desktop, karena pengguna cenderung menggunakan perangkat *mobile* dalam situasi yang lebih dinamis.

5. Memanfaatkan fitur – fitur Unik Perangkat *Mobile*

Aplikasi *mobile* dibuat untuk memanfaatkan fitur-fitur unik perangkat seluler seperti GPS, kamera, sensor, notifikasi push, dan lain-lain. Fitur-fitur ini memungkinkan aplikasi memberikan layanan yang lebih kaya dan kontekstual berdasarkan lokasi, kebiasaan, atau data real-time dari pengguna.

6. Menyediakan Layanan Secara Personal dan Real-Time

Aplikasi *mobile* memungkinkan pengguna mendapatkan layanan yang dipersonalisasi dan responsif dalam waktu nyata. Dengan mengandalkan data

pengguna dan analitik, aplikasi dapat menawarkan konten yang disesuaikan dengan preferensi atau perilaku pengguna, sehingga meningkatkan relevansi dan nilai bagi mereka.

7. Memfasilitasi Pengumpulan dan Analisa Data

Aplikasi *mobile* juga sering digunakan oleh perusahaan untuk mengumpulkan data tentang perilaku pengguna. Data ini kemudian dapat dianalisis untuk memahami preferensi pengguna, memperbaiki produk atau layanan, dan mengoptimalkan pengalaman pengguna.

Secara keseluruhan, tujuan aplikasi *mobile* adalah untuk memberikan aksesibilitas yang lebih baik, meningkatkan produktivitas, mendukung interaksi dan komunikasi, serta memanfaatkan teknologi *mobile* untuk menciptakan pengalaman yang lebih personal, efisien, dan bermanfaat bagi pengguna. Dalam konteks bisnis, aplikasi *mobile* juga bertujuan untuk memperkuat pemasaran, meningkatkan loyalitas pelanggan, serta menyediakan solusi yang relevan dan terukur bagi pengguna. Berdasarkan Bahasa Pemrograman

1. *Server side* adalah web yang menggunakan bahasa pemrograman yang bergantung pada kesediaan server seperti PHP, ASP, dan sebagainya. Jika tidak ada server, maka *website* yang dibangun menggunakan bahasa pemrograman tidak dapat berfungsi sebagaimana mestinya.
2. *Client side* adalah *website* yang tidak membutuhkan *server* dalam menjalankannya, *website* ini cukup di akses melalui *browser*.

2.2.3 Fungsi Aplikasi *Mobile*

Aplikasi *mobile* memiliki berbagai fungsi yang bergantung pada jenis, tujuan, dan sektor penggunaan. Secara umum, fungsi aplikasi *mobile* dirancang untuk memudahkan pengguna dalam melakukan berbagai aktivitas melalui perangkat seluler. Berikut adalah beberapa fungsi utama aplikasi *mobile* yaitu: (Kamilla et al., 2024)

1. Mempermudah Akses Informasi

Aplikasi *mobile* berfungsi untuk memberikan akses cepat dan mudah ke berbagai jenis informasi, seperti berita, cuaca, peta, atau data pribadi. Pengguna dapat mengakses informasi ini kapan saja dan di mana saja tanpa harus menggunakan perangkat komputer.

2. Meningkatkan Produktivitas

Aplikasi *mobile* membantu meningkatkan produktivitas individu maupun organisasi dengan menyediakan alat untuk manajemen tugas, pengingat, kalender, pengeditan dokumen, dan kolaborasi tim. Fungsi ini memungkinkan pengguna untuk menyelesaikan pekerjaan atau tugas pribadi dengan lebih efisien

3. Meningkatkan Pelayanan Pelanggan

Banyak perusahaan menggunakan aplikasi *mobile* untuk meningkatkan pengalaman pelanggan dengan memberikan layanan langsung, seperti pembelian produk, pengelolaan akun, pemesanan, dan pengiriman layanan pelanggan. Aplikasi ini memfasilitasi interaksi yang lebih mudah antara bisnis dan pelanggan.

4. Mendukung Pengumpulan Data dan Analisis

Aplikasi *mobile* juga memiliki fungsi untuk mengumpulkan data dari pengguna, seperti preferensi, kebiasaan, atau pola penggunaan. Data ini kemudian dapat dianalisis untuk memberikan wawasan yang lebih baik kepada perusahaan atau pengembang untuk meningkatkan aplikasi atau layanan

Fungsi utama aplikasi *mobile* adalah untuk memudahkan kehidupan pengguna dengan memberikan akses ke informasi, layanan, dan alat produktivitas yang bisa diakses kapan saja dan di mana saja. Selain itu, aplikasi *mobile* meningkatkan komunikasi, hiburan, transaksi bisnis, dan bahkan pengelolaan kesehatan, pendidikan, dan keuangan, membuatnya sangat penting dalam kehidupan modern.

2.3 Android eclips

Android Eclipse adalah istilah yang merujuk pada penggunaan Eclipse sebagai Integrated Development Environment (IDE) untuk mengembangkan aplikasi berbasis Android. Eclipse sendiri merupakan IDE bersifat open source dan multiplatform, yang berarti dapat dijalankan pada berbagai sistem operasi, seperti Windows, Linux, maupun macOS. (Machmudi et al., 2024).

2.3.1 Sejarah dan Peran Eclipse dalam Pengembangan Android

1. Sebelum adanya Android Studio, Eclipse adalah IDE utama yang digunakan oleh para developer Android untuk membuat berbagai jenis aplikasi, mulai dari permainan, aplikasi kamus, hingga aplikasi kontrol perangkat elektronik.
2. Google mendukung Eclipse dengan menyediakan plugin khusus bernama Android Developer Tools (ADT). Plugin ini memungkinkan Eclipse untuk melakukan kompilasi, pengujian, dan debugging aplikasi Android secara langsung.
3. Eclipse tetap digunakan oleh sebagian pengembang karena ringan, gratis, dan open source. Namun, sejak Google merilis Android Studio sebagai IDE resmi, dukungan untuk Eclipse mulai dihentikan, sehingga fitur-fitur terbaru Android tidak lagi tersedia di Eclipse.

2.3.2 Fitur dan Arsitektur Eclipse

1. Eclipse mendukung pengembangan aplikasi menggunakan berbagai bahasa pemrograman, terutama Java, namun juga mendukung C/C++, Python, dan lain-lain melalui sistem plugin.
2. Eclipse memiliki arsitektur berbasis plugin yang memungkinkan pengembang menambah atau mengurangi fitur sesuai kebutuhan. Setiap plugin dapat fokus pada tugas tertentu, seperti kompilasi, debugging, atau desain antarmuka pengguna.
3. Lingkungan kerja Eclipse terdiri dari beberapa komponen utama seperti Project Explorer (untuk navigasi file dan folder proyek), area editor (untuk menulis kode), dan console debugging (untuk memantau proses debugging).

2.3.3 Struktur Proyek Android di Eclipse

Dalam pengembangan aplikasi Android menggunakan Eclipse, struktur proyek biasanya terdiri dari beberapa folder penting:

1. **src/**: Tempat file-file Java utama, termasuk kelas Activity yang menjadi logika dasar aplikasi Android.
2. **gen/**: Berisi file Java yang dihasilkan secara otomatis (generated files), seperti R.java yang berfungsi untuk menghubungkan resource aplikasi.
3. **res/**: Menyimpan resource aplikasi seperti layout, gambar, dan string.
4. **AndroidManifest.xml**: File inti yang mendeskripsikan komponen aplikasi, permission, dan konfigurasi lainnya.

2.4 Java

Java Android merujuk pada penggunaan bahasa pemrograman Java sebagai fondasi utama dalam pengembangan aplikasi Android. Sejak awal kemunculan Android, Java telah menjadi bahasa utama untuk menulis kode aplikasi yang berjalan di atas sistem operasi Android. (Bagus Pribadi Putra & Siti Aliyah, 2023).

2.4.1 Peran Java dalam Pengembangan Android

1. Bahasa Utama: Java digunakan untuk membuat berbagai komponen aplikasi Android, seperti Activity, Service, Broadcast Receiver, dan Content Provider.
2. Dukungan SDK: Android Software Development Kit (SDK) menyediakan API dan alat bantu yang memungkinkan pengembangan aplikasi menggunakan Java.

3. **Kompatibilitas:** Java didukung penuh oleh Android Studio, IDE resmi pengembangan Android, sehingga pengembang dapat menulis, menguji, dan menjalankan kode Java secara langsung.

2.4.2 Konsep Dasar Java untuk Android

1. **Sintaks Dasar:** Pengembang harus memahami variabel, tipe data (int, float, char, boolean), kontrol alur (if, for, while), kelas, dan objek sebagai dasar pemrograman Java.
2. **OOP (Object-Oriented Programming):** Java menerapkan konsep OOP seperti enkapsulasi, pewarisan, dan polimorfisme, yang sangat penting dalam struktur aplikasi Android.
3. **Exception Handling & Multithreading:** Penanganan error dan proses paralel sangat dibutuhkan untuk aplikasi Android yang responsif dan stabil.

2.4.3 Struktur Program Java di Android

1. **Activity:** Setiap tampilan atau layar aplikasi biasanya diwakili oleh sebuah kelas Java yang disebut Activity. Activity menangani interaksi pengguna dan logika aplikasi.
2. **Intent:** Objek Java yang digunakan untuk berpindah antar Activity atau mengirim data antar komponen aplikasi.
3. **Resource Management:** Java digunakan untuk mengakses dan mengelola resource aplikasi, seperti layout XML, gambar, dan string.

2.5 NOX

Nox atau NoxPlayer adalah sebuah aplikasi emulator Android yang digunakan untuk menjalankan aplikasi dan game Android di komputer (PC atau laptop). Dengan Nox, pengguna dapat mensimulasikan perangkat Android secara virtual di sistem operasi Windows atau Mac, sehingga aplikasi Android bisa dijalankan seperti di perangkat aslinya.

2.5.1 Kegunaan Utama Nox

1. Menjalankan aplikasi/game Android di PC: Nox memungkinkan pengguna menginstal dan menggunakan aplikasi atau game Android di komputer, sehingga dapat menikmati layar yang lebih besar dan kontrol menggunakan keyboard serta mouse.
2. Multi-Drive: Fitur ini memungkinkan pengguna menjalankan beberapa emulator Android secara bersamaan di satu PC, sehingga bisa menggunakan banyak akun atau aplikasi sekaligus.
3. Testing aplikasi: Developer aplikasi Android dapat menggunakan Nox untuk menguji aplikasi mereka tanpa perlu perangkat fisik.
4. Fitur tambahan: Nox menyediakan fitur seperti screenshot, perekaman video, pengaturan volume, import/export file, dan mode fullscreen.
5. Stabil dan cepat: Nox dikenal responsif, tidak mudah freeze, dan mendukung mode performa tinggi untuk pengalaman bermain game yang optimal.

2.6 Sistem Transaksi

Sistem transaksi adalah sistem yang dirancang untuk mencatat, memproses, dan menyimpan aktivitas transaksi yang dilakukan dalam suatu organisasi atau bisnis. Transaksi di sini bisa berupa pertukaran barang, jasa, atau informasi biasanya berkaitan dengan keuangan, tetapi juga bisa non-keuangan. Sistem ini sering disebut juga sebagai Transaction Processing System (TPS) dalam dunia IT.(Maharani, 2023).

2.6.1 Tujuan Sistem Transaksi

1. Mencatat semua aktivitas transaksi dengan akurat
2. Memastikan kecepatan dan efisiensi dalam pemrosesan
3. Menyediakan data yang bisa digunakan untuk pelaporan dan analisis
4. Meminimalisir kesalahan manusia (human error)
5. Memberikan riwayat dan jejak transaksi yang bisa ditelusuri.

2.6.2 Karakteristik Sistem Transaksi

1. Volume Tinggi & Repetitif

Sistem menangani banyak transaksi setiap hari (misalnya: penjualan, pembayaran, input data pasien, dll).

2. Real-time atau Batch

Bisa berjalan langsung saat transaksi dilakukan (**real-time**) atau dikumpulkan dan diproses kemudian (**batch**).

3. Terstruktur & Konsisten

Data yang diinput harus mengikuti format dan alur yang ditentukan.

4. **Reliabilitas Tinggi**

Harus stabil dan bisa diandalkan karena menyangkut data penting.

5. **Keamanan Data**

Transaksi harus aman, apalagi jika menyangkut uang atau data pribadi.

2.7 **Algoritma Greedy**

Algoritma greedy adalah algoritma yang biasa digunakan dalam menyelesaikan masalah optimasi. Algoritma greedy juga sebagai metode populer dalam memecahkan persoalan optimasi, dimanfaatkan untuk mencari nilai nominal terdekat pada pecahan mata uang, sehingga jumlah uang yang akan ditampilkan pada layar layanan kesehatan sesuai dengan total pembelian. Dengan implementasi sistem ini, layanan kesehatan dapat secara otomatis memberikan pecahan uang kembalian yang tepat, mengurangi human error, dan menghindari kecurangan karyawan. Batasan masalah dalam penelitian ini meliputi:(Sunata et al., 2024)

1. Penentuan pecahan kembalian uang pada mesin layanan kesehatan.
2. Memastikan nilai total dari himpunan pecahan yang dipilih tidak melebihi jumlah uang yang harus dikembalikan.
3. Penerapan algoritme greedy untuk penyelesaian masalah.
4. Output berupa jumlah pecahan uang yang dikeluarkan dengan nominalnya.
5. Interaksi pengguna melalui input total pembelian, total uang yang diterima, dan eksperimen dengan kombinasi pecahan untuk menghasilkan total kembalian yang tepat.

Algoritma Greedy (algoritma rakus) adalah salah satu metode penyelesaian masalah komputasi dengan cara mengambil keputusan yang terbaik (optimal) pada setiap langkah lokal, dengan harapan bahwa keputusan tersebut akan menghasilkan solusi yang optimal secara global.

Dengan kata lain, algoritma Greedy selalu memilih langkah yang **paling** menguntungkan saat itu juga, tanpa mempertimbangkan konsekuensi jangka panjang atau solusi keseluruhan.

Namun, penting dicatat bahwa algoritma Greedy tidak selalu menghasilkan solusi optimal untuk semua jenis masalah. Algoritma ini hanya optimal untuk masalah-masalah yang memiliki struktur tertentu, misalnya optimal substructure dan greedy choice property.

2.7.1 Elemen-Elemen Kunci Algoritma Greedy

Algoritma greedy memiliki lima elemen kunci: himpunan kandidat, himpunan solusi, fungsi seleksi, fungsi kelayakan, dan fungsi tujuan. Himpunan kandidat adalah kumpulan semua kemungkinan solusi, himpunan solusi adalah kumpulan solusi yang telah dipilih, fungsi seleksi menentukan pilihan terbaik di setiap langkah, fungsi kelayakan memastikan pilihan yang dibuat adalah valid, dan fungsi tujuan adalah kriteria untuk mengukur kualitas solusi.

Berikut adalah penjelasan lebih rinci mengenai elemen-elemen algoritma greedy:

1. Himpunan Kandidat (Candidate Set):

Ini adalah kumpulan semua kemungkinan pilihan atau solusi yang tersedia untuk dipertimbangkan.

2. Himpunan Solusi (Solution Set):

Ini adalah himpunan yang berisi solusi yang telah dipilih oleh algoritma greedy. Pada setiap langkah, algoritma akan memilih elemen dari himpunan kandidat dan menambahkannya ke himpunan solusi.

3. Fungsi Seleksi (Selection Function):

Fungsi ini digunakan untuk memilih elemen terbaik dari himpunan kandidat pada setiap langkah. Pilihan ini didasarkan pada kriteria tertentu yang diharapkan dapat mengarah pada solusi optimal.

4. Fungsi Kelayakan (Feasibility Function):

Fungsi ini memeriksa apakah pilihan yang dibuat oleh fungsi seleksi masih memenuhi batasan atau aturan yang berlaku. Jika pilihan tersebut melanggar batasan, maka pilihan tersebut akan ditolak.

5. Fungsi Tujuan (Objective Function):

Fungsi ini digunakan untuk mengukur kualitas solusi yang dihasilkan oleh algoritma. Tujuan dari algoritma greedy adalah untuk memaksimalkan atau meminimalkan nilai dari fungsi tujuan.

Dalam setiap langkah algoritma greedy, elemen dipilih dari himpunan kandidat berdasarkan fungsi seleksi. Elemen yang dipilih kemudian diperiksa oleh fungsi kelayakan untuk memastikan bahwa elemen tersebut tidak melanggar batasan. Jika elemen tersebut layak, maka akan ditambahkan ke himpunan solusi. Proses ini berulang hingga solusi yang optimal ditemukan

2.7.2 Prinsip dan Cara Kerja Algoritma Greedy pada Sistem Layanan kesehatan

Pada setiap langkah, algoritma greedy akan selalu memilih pecahan uang terbesar yang masih kurang dari atau sama dengan sisa jumlah kembalian yang harus diberikan. Proses ini diulangi hingga seluruh jumlah kembalian terpenuhi, sehingga jumlah lembar/koin yang diberikan menjadi minimum. (Lakutu et al., 2023)

Rumusan Algoritma Greedy untuk Penukaran Uang Kembalian

Secara umum, rumusan algoritma greedy untuk masalah kembalian di layanan kesehatan adalah sebagai berikut:

1. Diberikan:

Daftar pecahan uang yang tersedia,
misal: $D = \{100000, 50000, 20000, 10000, 5000, 2000, 1000, 500, 100\}$

Jumlah kembalian yang harus diberikan: KK .

2. Langkah-langkah:

- a. Urutkan pecahan uang dari yang terbesar ke terkecil.
- b. Untuk setiap pecahan dd dalam DD :
- c. Selama $K \geq dd$, ambil satu lembar/koin pecahan dd , kurangi KK sebesar dd , dan catat jumlah pecahan yang diambil.
- d. Ulangi hingga $K=0$.

2.7.3 Pseudocode Algoritma Greedy pada system transaksi

Input:

DaftarBarang[] // array berisi harga tiap barang yang dibeli
 UangDibayar // jumlah uang yang diserahkan pelanggan
 Pecahan[] // array pecahan uang,urut dari terbesar ke terkecil

Output:

TotalBelanja // total harga semua barang
 Kembalian // uang kembalian yang harus diberikan
 JumlahPecahan[] // jumlah tiap pecahan uang kembalian yang diberikan

```
// Hitung total belanja
TotalBelanja = 0
for i = 0 to DaftarBarang.length - 1 do
    TotalBelanja = TotalBelanja + DaftarBarang[i]
end for

// Hitung kembalian
Kembalian = UangDibayar - TotalBelanja
if Kembalian < 0 then
    Output "Uang yang dibayar kurang"
    stop
end if

// Hitung pecahan kembalian menggunakan algoritma greedy
for i = 0 to Pecahan.length - 1 do
    JumlahPecahan[i] = 0
```

```
while Kembalian >= Pecahan[i] do
    Kembalian = Kembalian - Pecahan[i]
    JumlahPecahan[i] = JumlahPecahan[i] + 1
end while

end for

if Kembalian > 0 then
    Output "Tidak dapat memberikan kembalian dengan pecahan yang tersedia"
else
    Output TotalBelanja
    Output UangDibayar
    Output JumlahPecahan untuk tiap pecahan uang kembalian
end if
```

Penjelasan:

1. TotalBelanja dihitung dari penjumlahan harga semua barang.
2. Kembalian adalah selisih antara uang yang dibayar dan total belanja.
3. Algoritma greedy memilih pecahan uang terbesar yang masih bisa digunakan untuk mengurangi kembalian, sehingga jumlah lembar/koin kembalian minimal.
4. Jika kembalian negatif, berarti uang yang dibayar kurang.
5. Jika setelah proses masih ada sisa kembalian, berarti pecahan uang yang tersedia tidak bisa memenuhi kembalian secara tepat.

Contoh Kasus

Misal, kembalian yang harus diberikan adalah Rp32.000, dan pecahan yang tersedia adalah Rp20.000, Rp10.000, Rp5.000, Rp2.000, Rp1.000.

- Pilih Rp20.000 (sisa: 12.000)
- Pilih Rp10.000 (sisa: 2.000)
- Pilih Rp2.000 (sisa: 0)

Total: 1 lembar Rp20.000, 1 lembar Rp10.000, 1 lembar Rp2.000 (total 3 lembar, solusi optimal).

2.8 SQL Lite

SQLite adalah sebuah software atau sistem manajemen relational database yang bersifat open-source, serverless, dan portabel yang digunakan untuk mempermudah akses dan mengelola penyimpanan data Anda. Software ini didesain untuk tujuan utama dalam hal kecepatan, kemudahan, portabel, dan dapat diandalkan ketika Anda butuh untuk menyimpan data hanya 1 kilobytes ataupun ratusan gigabytes.

Penggunanya dapat dengan mudah menggunakan SQLite tanpa perlu melakukan instalasi environment atau konfigurasi apapun karena SQLite merupakan stand-alone software yang memang didesain untuk disematkan (embedded) ke dalam sebuah aplikasi. Selain itu, beberapa produk sistem manajemen relational database yang cukup populer lainnya adalah MySQL, Oracle Database, PostgreSQL memiliki ukuran yang jauh lebih besar dibandingkan SQLite.

Disamping itu SQLite juga tidak menggunakan model arsitektur client-server seperti MySQL dan PostgreSQL karena seluruh programnya berisi library Bahasa pemrograman C yang menyematkan database ke dalam sebuah aplikasi. Ini berarti Anda tidak membutuhkan konfigurasi server yang rumit untuk dapat menjalankan sistem manajemen database pada aplikasi Anda. (Pasaribu & Azhari, 2024).

2.8.1 Cara Kerja SQL Lite

Jika dibandingkan dengan produk relational database lainnya yang menerapkan model client-server, SQLite bekerja dengan menyimpan data Anda di satu file lintas platform. Dengan menggunakan model client-server berarti Anda harus menyiapkan segala hal yang diperlukan seperti dedicated server, melakukan instalasi environment yang dibutuhkan, melakukan testing terhadap performa database Anda pada sisi client, dan masih banyak yang lainnya.

Namun, dengan SQLite akan sangat mudah hanya dengan menautkan library yang dibutuhkan dan membuat file regular baru sehingga Anda akan langsung dapat menjalankan database. Kemudahan ini akan membuat Anda untuk fokus mengerjakan pekerjaan administrasi data, ataupun melakukan analisis terhadap data yang Anda miliki untuk memperoleh informasi yang Anda butuhkan.

Ketika Anda menyimpan data pada SQLite, maka dibutuhkan sebuah bahasa untuk melakukan query atau perintah-perintah seperti membuat suatu tabel, mengedit value dari kolom tertentu, menambahkan beberapa data baru, menghapus data, ataupun pemrosesan data lainnya. Bahasa query ini adalah SQL (Structured

Query Language). SQL menjadi bahasa yang populer terutama untuk relational database karena dapat mengolah ataupun memanipulasi data dengan sangat cepat. Disini SQL digunakan untuk mengolah dan memanipulasi data seperti menambahkan data, menghapus data, melakukan update terhadap data, menampilkan data, hingga melakukan analisa data. Penggunaan SQL sangat praktis dan efisien dalam mengelola data cepat terlebih lagi jika Anda mempunyai kendala dan keterbatasan. Disamping itu, Anda tidak akan pernah khawatir dengan pekerjaan tambahan seperti mengurus infrastruktur atau mengeluarkan biaya untuk server pada aplikasi.

2.8.2 Kelebihan SQL Lite

1. Ringan

Dengan ukuran yang sangat kecil sekitar 500kb dan kemudahan penggunaan yang ditujukan untuk embedded system menjadikan SQLite database tidak akan memberatkan aplikasi Anda. Anda juga tidak perlu menunggu waktu download yang lebih lama dan ini tidak akan menghabiskan kuota data Anda karena SQLite ini ringan dan hemat dalam menggunakan resource.

2. Tidak memerlukan proses instalasi

Anda dapat mudah menjalankan database dan mengetikkan perintah SQL Anda karena SQLite bersifat portable sehingga tidak perlu melakukan instalasi apapun saat ingin menggunakannya. Hanya dalam beberapa klik dan kemudian database telah siap digunakan oleh Anda.

3. Bersifat Open Source

Karena bersifat open-source, ini artinya SQLite tidak membutuhkan lisensi apapun atau gratis untuk menggunakannya. Bagi Anda yang sedang mengembangkan sebuah project, ini akan menghemat pengeluaran biaya Anda sehingga dapat dimanfaatkan untuk meningkatkan pendapatan Anda. Dapat digunakan di platform yang berbeda-beda. SQLite dapat dijalankan diberbagai sistem operasi seperti Windows, macOS, Linux, Android dan sebagainya. Ini akan memudahkan Anda jika project anda ada di cross platform.

4. Membantu dalam melakukan analisa data

Menyimpan data di database, maka Anda juga memungkinkan untuk melakukan menganalisa data yang Anda miliki sehingga mendapatkan informasi yang Anda butuhkan. Dengan mengetikan beberapa query SQL dan menjalankannya, Anda akan memperoleh informasi dalam jangka waktu yang sangat cepat serta data ini dapat diakses oleh Anda dengan mudah. Melakukan analisa seperti kenaikan penjualan beberapa bulan, top 3 product teratas Anda, ranking pelanggan Anda, dan sebagainya tidak akan membuang banyak waktu Anda lebih banyak.

5. Menjaga keamanan data dengan baik

Anda dapat melakukan backup data di database SQLite sehingga jika terdapat kejadian tidak terduga di masa mendatang seperti error database atau serangan siber dari oknum yang tidak bertanggung jawab, maka Anda tidak perlu mengkhawatirkan hal-hal yang berkaitan dengan keamanan data.

2.8.3 Kekurangan SQL Lite

Layaknya dua sisi mata uang dan dengan beragam kemudahan dan kecepatan ditawarkan oleh SQLite, terdapat beberapa kekurangan dalam software SQLite yang harus Anda pahami seperti:

1. Hanya dapat menangani low-medium HTTP request
2. Ukuran penyimpanan terbatas, tergantung memori device yg terdapat di dalam aplikasi tersebut.
3. Jika pindah device maka database juga akan hilang


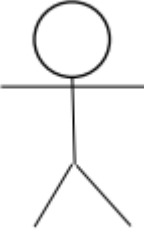


2.9 Unified Modeling Language (UML)



Unified Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah software yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam blue print yang didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem.

2.9.1 Use Case

Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam use case terdapat actor yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem. Berikut adalah simbol-simbol yang terdapat pada *usecase diagram*.

Tabel 2.1 Simbol-Simbol Usecase Diagram



Gambar	Nama	Keterangan
	<i>Usecase</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dengan aktor yang dinyatakan.
	<i>Actor</i>	Merupakan abstraksi dari <i>user</i> sistem yang mengoperasikan fungsi dari beberapa modul sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian peran dan tugas-tugas yang sesuai konteks target sistem, tetapi tidak memiliki kontrol penuh terhadap <i>usecase</i> .
	<i>Association</i>	Asosiasi antara aktor dan <i>usecase</i> , digambarkan dengan garis. Tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	<i>Association</i>	Asosiasi antara aktor dan <i>usecase</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor



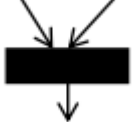

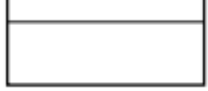
		berinteraksi secara pasif dengan sistem.
	<i>Include</i>	<i>Usecase</i> oleh <i>usecase</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i>	Merupakan perluasan dari <i>usecase</i> lain jika kondisi atau syarat terpenuhi.

2.9.2 Activity Diagram

Merupakan gambaran alir dari aktivitas-aktivitas didalam sistem yang berjalan. Berikut adalah simbol-simbol yang digunakan dalam *activity diagram*, dapat dilihat pada Tabel 2.2

Tabel 2.2 Simbol-Simbol Activity Diagram

Gambar	Nama	Keterangan
	<i>Start Point</i>	merupakan simbol yang diletakkan pada pojok kiri atas dan merupakan awal suatu aktivitas.
	<i>End Point</i>	merupakan simbol yang diletakkan pada pojok bawah dan merupakan berakhirnya suatu aktivitas.

	<i>Activities</i>	menggambarkan suatu proses / kegiatan.
	<i>Fork</i>	percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	<i>Join</i>	(penggabungan) atau <i>join</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Point</i>	menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
	<i>Swimlane</i>	pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

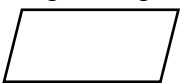



2.9.3 Flowchart

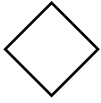


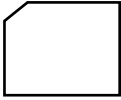
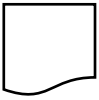




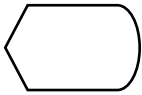
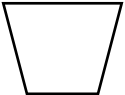
Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah – langkah intruksi yang berurutan dalam sistem. Seorang analis sistem menggunakan flowchart sebagai bukti dokumentasi yang menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada programmer.

Dengan begitu, flowchart dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. pada dasarnya, flowchart digambarkan dengan menggunakan simbol – simbol. Setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan suatu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung.

Dengan adanya flowchart, setiap urutan proses dapat digambarkan menjadi lebih jelas. Selain itu, ketika ada penambahan proses baru dapat dilakukan dengan mudah menggunakan flowchart ini. Setelah proses membuat flowchart selesai, maka giliran programmer yang akan menerjemahkan desain logis tersebut kedalam bentuk program dengan berbagai bahasa pemograman yang telah disepakati (Rosaly & Prasetyo, 2020).

Tabel 2.3 Simbol-Simbol Flowchart

Simbol	Arti
<p><i>Input/output</i></p> 	Mempresentasikan input data atau output data yang diproses atau informasi
<p>Proses</p> 	Mempresentasikan Operasi
<p>Penghubung</p> 	Keluar ke atau masuk dari bagian lain flowchart khususnya halaman yang sama
<p>Anak Panah</p> 	Mempresentasikan alur kerja

Keputusan 	Keputusan dalam program
<i>Predefined</i> 	Rincian operasi berada di tempat lain
Preparation 	Pemberian harga awal
<i>Punched Card</i> 	Input / Output yang menggunakan kartu berlubang
Dokumen 	Input / Output dalam format yang dicetak
Magnetic Drum 	Input / Output yang menggunakan Drum Magnetic
Magnetic Disk 	Input / Output yang menggunakan Disk Magnetic
Punched Tape 	Input / Output yang menggunakan pita kertas berlubang
Manual Input 	Input yang dimasukkan secara manual dari keyboard
Display 	Output yang ditampilkan pada terminal
Manual Operation 	Operasi Manual