BAB II

TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

2.1.1 Sejarah

PT Pangan Rahmat Buana merupakan perusahaan yang bergerak di bidang pembuatan roti yang berdiri pada tanggal 23 Januari 2001, dan berlokasi di Sentul - Bogor. PT. Pangan Rahmat Buana mampu memproduksi roti berkualitas tinggi menggunakan tepung istimewa yang diproduksi oleh PT Sriboga Raturaya Flourmill yang sudah menerima pengakuan dunia dari UNICEF pada tahun 1999 sebagai pelopor yang komprehensif dalam bidang fortifikasi tepung terigu. PT. Pangan Rahmat Buana memiliki brand utama roti Le Gitt sebagai brand massal konsumen, juga memproduksi roti untuk konsumsi institusi ataupun yang ditujukan untuk private label seperti roti hotdog, burger, maupun breadcrumb.

Visi :

Menyediakan penganan roti yang enak, berkualitas dan kaya nutrisi untuk keluarga Indonesia.

Misi :

1. Membangun jaringan distribusi yang kuat untuk pemenuhan produk setiap saat
2. Memproduksi roti yang kaya manfaat nutrisi
3. Memberikan kontribusi dan kerjasama yang baik untuk semua stakeholders

2.1.2 Logo Instansi

Gambar 2.1 Logo PT Pangan Rahmat Buana

Gambar 2.2 Logo Brand Le Gitt PT Pangan Rahmat Buana

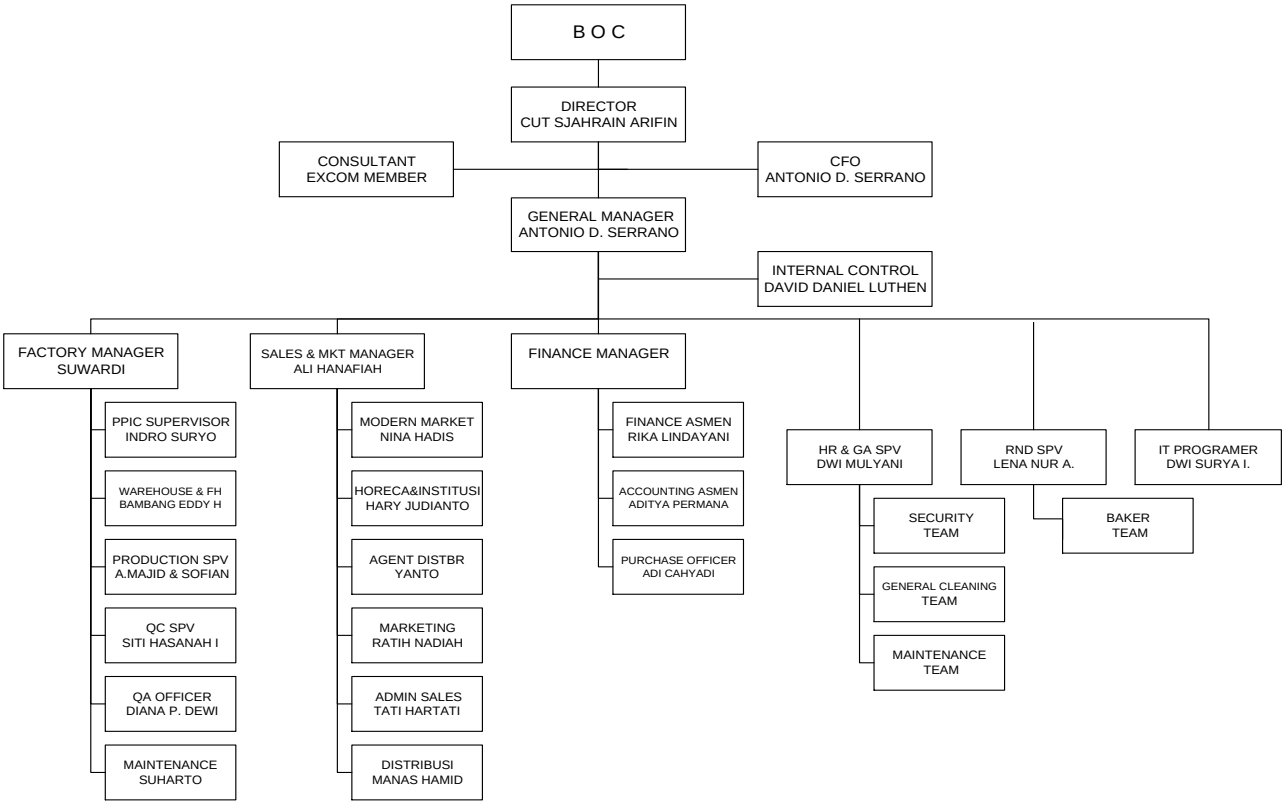
2.1.3 Badan Hukum

PT Pangan Rahmat Buana adalah suatu badan usaha perseroan yang didirikan pada tanggal 23 januari 2001 dengan SIUP No. 8050/09-04/PB/XI/95. Perusahaan ini didirikan berdasarkan Undang – Undang Negara Republik Indonesia dengan akta No. 8 tertanggal 25 Oktober 2001.

2.1.4 Struktur Organisasi dan Job Description

PT. Pangan Rahmat Buana dipimpin oleh seorang direktur. Tugas dan tanggung jawab seorang Direktur yaitu mengelola jalannya kegiatan operasional perusahaan. Tugas seoarang Direktur dibantu GM (General Manager). GM merupakan pimpinan manajer dari semua divisi yang berada si perusahaan tersebut. GM ini bertugas mengawasi dan menjalankan kegiatan manajemen dalam perusahaan yang dibantu oleh manajer lain dan membawahi Purchasing Order, Factory Manager, F & A Manger, Sales *Marketing* Manager, HRD Supervisor dan GA Asisten Manager. Berikutb adalah Struktur organisasi dan Job Description manajer PT. Pangan Rahmat Buana.

2.1.4.1 Struktur Organisasi



Gambar 2.3 Struktur Organisasi

2.1.4.2 Job Description

Tabel 1.1 Job Description

Jenis-Jenis Manager	**Tugas dan Tanggung Jawab**
Purchasing Order Manager	Pengadaan barang-barang untuk perusahaan, baik itu keperluan produksi seperti bahan baku, bahan penunjang, mesin dan peralatan maupu untuk keperluan perusahaan lainnya.
Factory Manager	Mengelola dan mengawasi semua kegiatan pabrik dalam proses produksi. Seorang Factory Manager membawahi Prduction Supervisor, PD Supervisor dan QC Supervisor.
Finance & Accounting Manager	Membawahi Finance Asissten Manager dan Cost Accounting Supervisor. F & A Mager ini bertanggung jawab untuk mengawasi keuangan dari penjualan, selain itu juga bertugas dalam menghitung stock opname terhadap bahan baku, biaya pajak, biaya produk-produk yang ditolak atau dikembalikan maupun biaya operasional umum.
Sales & *Marketing* Manager	Manager mambawahi Head Sales and Head *Marketing*.
HRD Supervisor	Membawahi IT staff dan HRD staff.
GA Asisten Manager	Membawahi koordinator Maintanance, koordinator kendaraan dan koordinator distribusi.

2.2 Landasan Teori

2.2.1 Pengertian Sistem

Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

2.2.2 Pengertian Informasi

Informasi adalah data yang diolah dan dibentuk menjadi lebih berguna dan lebih berarti bagi yang menerimanya. Informasi merupakan pengumpulan dan pengolahan data untuk memberikan keterangan atau pengetahuan.

2.2.3 Pengertian Sistem Informasi

Sistem Informasi (SI) adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algoritmik, data, dan teknologi. Dalam pengertian ini, istilah ini digunakan untuk merujuk tidak hanya pada penggunaan organisasi teknologi informasi dan komunikasi (TIK), tetapi juga untuk cara di mana orang berinteraksi dengan teknologi ini dalam mendukung proses bisnis.

2.2.4 Pengertian Inventory

Inventory atau persediaan merupakan bahan atau barang yang disimpan yang akan digunakan untuk memenuhi tujuan tertentu, misalnya untuk proses produksi atau perakitan, untuk dijual kembali, dan untuk suku cadang dari suatu peralatan. Inventory dapat berupa bahan mentah, bahan pembantu, barang dalam proses, barang jadi, ataupun suku cadang.

2.2.5 Pengertian Inventory Control

Inventory Control adalah pengawasan pasokan, penyimpanan dan aksesibilitas item dalam rangka untuk memastikan pasokan yang cukup tanpa kelebihan pasokan yang berlebihan. Hal ini juga dapat disebut sebagai kontrol internal - prosedur akuntansi atau

sistem yang dirancang untuk meningkatkan efisiensi atau menjamin pelaksanaan kebijakan atau menjaga aset atau menghindari penipuan dan kesalahan dll.

Inventory control dapat merujuk kepada:

- Dalam ilmu ekonomi, pengendalian persediaan, yang bertujuan untuk mengurangi biaya overhead tanpa menyakiti penjualan.
- Dalam bidang pencegahan kerugian, sistem yang dirancang untuk memperkenalkan hambatan teknis pencurian.

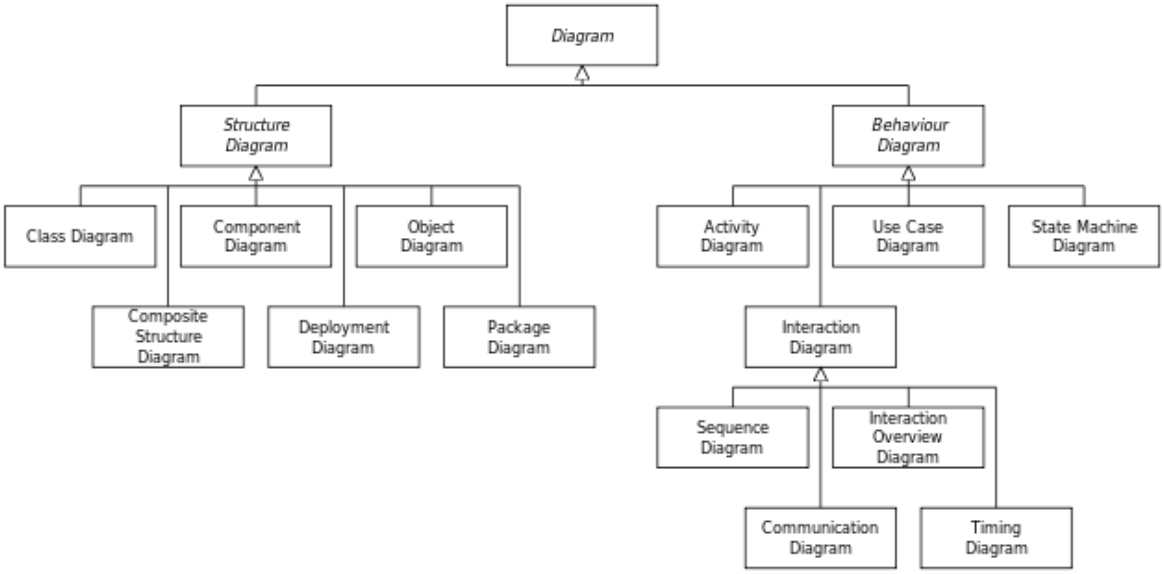
2.2.6 Pengertian Sistem Inventory Control

Sistem Inventory Control adalah suatu proses untuk mengelola dan menemukan benda atau bahan. Dalam penggunaan umum, istilah ini juga dapat merujuk hanya komponen perangkat lunak. Sistem pengendalian persediaan adalah cara sistematis menangani aliran material yang akan bermanfaat bagi industri dan membantu sistem inventory.

Sebuah *Sistem Inventory Control* dapat digunakan untuk mengotomatisasi proses penjualan pemenuhan order. Sistem seperti berisi daftar untuk diisi, dan kemudian meminta para pekerja untuk memilih item yang diperlukan, dan menyediakan mereka dengan kemasan dan pengiriman, Sebuah sistem persediaan juga mengelola input dan output dari materi bahan baku maupun produk jadi.

2.2.7 Pengertian UML

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek (*OOP*) oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.



Gambar 2.4 Model Struktur Pemodelan Diagram UML

Berikut 4 macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi object atau *object oriented programming* (OOP) , yaitu *use case diagram, sequence diagram, collaboration diagram,* dan *class diagram.*

1. *Use Case Diagram*

   Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Use case diagram terdiri atas diagram untuk use case dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. Use case merepresentasikan operasi-operasi yang dilakukan oleh *actor*. Use case digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke use case.

2. *Sequence Diagram*

   Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

3. *Collaboration Diagram*

Collaboration diagram dipakai untuk memodelkan interaksi antar object di dalam sistem. Berbeda dengan sequence diagram yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, collaboration diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh object.

4. *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

2.2.7.1 Tujuan UML diantaranya adalah untuk :

- Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti umum.
- Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- Menyatukan praktik-praktik terbaik (*best practise*) yang terdapat dalam pemodelan.

2.2.8 Pengertian StarUML

StarUML adalah software permodelan yang mendukung UML (*Unified Modeling Language*). Berdasarkan pada UML version 1.4 dan dilengkapi 11 macam diagram yang berbeda, mendukung notasi UML 2.0 dan juga mendukung pendekatan MDA (*Model Driven Architecture*) dengan dukungan konsep UML. StarUML dapat memaksimalkan produktivitas dan kualitas dari suatu software project.

2.2.9 Pengertian OOAD (*Object-Oriented Analysis and Design*)

Object-Oriented Analysis and Design (OOAD) adalah metode analisis yang memeriksa *requirements* dari sudut pandang kelas - kelas dan objek yang ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek system atau subsistem.OOAD merupakan

cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek,yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas.

2.2.9.1 Keuntungan Menggunakan *Object-Oriented Analysis and Design* (OOAD)

1. Meningkatkan produktivitas

   Karena kelas dan object yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).
2. Kecepatan pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.
3. Kemudahan pemeliharaan

   Karena dengan model objek , pola – pola yangcenderung tetap dan stabil dapat dipisahkan dan pola – pola yang mungkin sering berubah – ubah.
4. Adanya Konsistensi

   Karena sifat pewarisaan dan penggunaan notasi yang sama pada saat analisis, perancangan, maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak

   Karena pendektan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempuyai sedikit kesalahan.

2.2.9.2 Konsep Dasar *Object-Oriented Analysis and Design* (OOAD)

OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memerika requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan. Sedangkan OOD adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

Berikut adalah beberapa konsep dasar dalam OOAD, yaitu :

1. Kelas (*class*) adalah kumpulan objek-objek dengan karakteristik yang sama.
2. Objek (*object*) adalah abstraksi dan sesuatun yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak.
3. Metode (*method*) adalah mekanisme internal objek yang mencerminkan perilaku objek tersebut.
4. Atribut (*attribute*) adalah nilai internal suatu objek yang mencerminkan karakteristik objek, kondisi sesaat, koneksi dengan objek lain dan identitas.
5. Abstraksi (*abstraction*) adalah prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.
6. Enkapsulasi (*Encapsulation*) yaitu proses menyembunyikan detail implementasi sebuah objek. Untuk mengakses data objek tersebut adalah melalui interface. Untuk berkomunikasi dengan objek digunakan message.
7. Pewarisan (*inheritance*) yaitu mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.
8. Antarmuka (*interface*) adalah pintu untuk mengakses service dari objek
9. Reusablility yaitu pemanfaat kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.
10. Generalisasi dan spesialisasi yaitu menunjukan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.
11. Komunikasi antar objek dengan *message* adalah permintaan agar objek menerima untuk membawa metode yang ditunjukkan oleh perilaku dan mengembalikan result dari aksi tersebut kepada objek pengirim (sender)
12. Polimorfisme (*polymorephism*) yaitu kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.
13. *Package* adalah sebuah container atau kemasan yang dapat digunkan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.2.9.3 Teknik Pemodelan dalam *Object-Oriented Analysis and Design* (OOAD)

A. Model Objek :
   - Model objek Menggambarkan struktur statis dari suatu objek dalam sistem dan relasinya.
   - Model objek berisi diagram objek. Diagram objek adalah graph dimana nodenya adalah kelas yang mempunyai relasi antar kelas.

B. Model Dinamik
   - Model dinamik menggambarkan aspek dari sistem yang berubah setiap saat.
   - Model dinamik dipergunakan untuk menyatakan aspek kontrol dari sistem.
   - Model dinamik berisi state diagram. State diagram adalah graph dimana nodenya adalah state dan arc adalah tarnsisi antara state yang disebabkan oleh event.

C. Model Fungsional
   - Model fungsional menggambrakan transformasi nilai data di dalam sistem.
   - Model fungsional berisi data flow diagram. DFD adalah suatu *graph* dimana *node*nya menyatakan proses dan *arc*nya adalah aliran data.

2.2.10 Pengertian *Wamp Server*

Wamp adalah sebuah aplikasi yang dapat menjadikan komputer kita menjadi sebuah server. Kegunaan *wamp server* ini untuk membuat jaringan local sendiri dalam artian kita dapat membuat website secara offline untuk masa percoban di komputer sendiri.

2.2.11 Pengertian *phpMyAdmin*

phpMyAdmin merupakan sebuah *tools* yang dikembangkan menggunakan bahasa *PHP* dan ditujukan untuk menangani administrasi *MySQL* melalui *world wide web*. *phpMyAdmin* mendukung berbagai operasi pada *MySQL*, seperti membuat (*create*) dan menghapus (*drop*) *database*, membuat, menghapus, atau mengubah (alter) tabel, menghapus, mengedit, atau menambahkan field, mengeksekusi berbagai perintah *SQL*, atau mengelola *key* pada *field*.

2.2.12 Pengertian MySQL

MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak Pembuat *database* yang bersifat terbuka atau *open source* dan berjalan disemua *platform* baik *Linux*

maupun *Windows*, MySQL merupakan program pengakses *database* yang bersifat *network* sehingga dapat digunakan untuk aplikasi *Multi User* (Pengguna Banyak).

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial.

2.2.13 Pengetian Java

Java adalah suatu teknologi di dunia software komputer, yang merupakan suatu bahasa pemrograman, dan sekaligus suatu platform. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi. Java mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++. Java merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan.

Sebagai bahasa pemrograman, Java dirancang menjadi handal dan aman. Java juga dirancang agar dapat dijalankan di semua platform. Dan juga dirancang untuk menghasilkan aplikasi–aplikasi dengan performansi yang terbaik, seperti aplikasi database Oracle 8i/9i yang core-nya dibangun menggunakan bahasa pemrograman Java. Sedangkan Java bersifat neutral architecture, karena Java Compiler yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitekture perangkat keras yang disebut sebagai Java Bytecode. Sebagai sebuah platform, Java terdiri atas dua bagian utama, yaitu:

- *Java Virtual Machine* (JVM).
- *Java Application Programming Interface* (Java API).

Sun membagi arsitektur Java membagi tiga bagian, yaitu:

- *Enterprise Java* (J2EE) untuk aplikasi berbasis web, aplikasi sistem tersebar dengan beraneka ragam klien dengan kompleksitas yang tinggi. Merupakan superset dari *Standard Java*.
- *Standard Java* (J2SE), ini adalah yang biasa dikenal sebagai bahasa Java.
- *Micro Java* (J2ME) merupakan subset dari J2SE dan salah satu aplikasinya yang banyak dipakai adalah untuk wireless device/ mobile device.

2.2.13.1 Kelebihan Java

Beberapa kelebihan bahasa pemrograman Java dibandingkan dengan bahasa pemrograman lain :

a. Multiplatform

Kelebihan utama dari Java ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / bytecode) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebanya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan bytecode tersebut.

b. OOP (Object Oriented Programming – Pemrograman Berorientasi Objek)

OOP artinya semua aspek yang terdapat di Java adalah Objek. Java merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut Object. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan Java sebagai salah satu bahasa pemograman termudah, bahkan untuk fungsi fungsi yang advance seperti komunikasi antara computer sekalipun.

c. Perpustakaan kelas yang lengkap

Java terkenal dengan kelengkapan library/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

d. Bergaya C++

Java memiliki sintaks seperti bahasa pemrograman (C++) sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas - universitas di Amerika juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.

e. Pengumpulan sampah otomatis

Java memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2.2.14 Pengertian NetBeans IDE (*Integrated Development Environment*)

NetBeans IDE adalah sebuah *open source* terintegrasi dimana lingkungan pengembangannya ditulis di Java menggunakan Platform NetBeans. NetBeans IDE mendukung pengembangan semua tipe aplikasi Java (Java SE, web, EJB, dan aplikasi mobile). Diantara fitur-fitur lainnya adalah Projek Sistem Berbasis Ant, version control (mensuport CVS, Subversion, Mercurial dan Clearcase) dan refactoring.

Semua fungsi dari IDE disediakan oleh modul-modul. Setiap modul menyediakan fungsi yang didefinisikan dengan baik, seperti dukungan untuk bahasa Java, editing, atau dukungan untuk sistem versi CVS dan SVN. NetBeans berisi semua modul yang diperlukan untuk pengembangan Java, mengijinkan pengguna untuk memulai bekerja dengan segera. Modul-modul ini juga mengijinkan NetBeans untuk dikembangkan. Fitur-fitur baru, seperti dukungan untuk bahasa pemrograman lainnya, dapat ditambahkan dengan menginstal modul tambahan. Sebagai contoh, Sun Java Studio Enterprise dan Sun Java Studio Creator dari Sun Microsystem semuanya berbasis pada NetBeans IDE.