

BAB II

DASAR TEORI

2.1 Gelombang Bunyi

Gelombang bunyi termasuk dalam gelombang longitudinal dimana arah rambatnya sama dengan arah getarnya. Gelombang bunyi selalu memerlukan zat perantara (media) untuk proses perambatannya sehingga selalu disertai getaran zat antara yang dilaluinya. Kecepatan rambat dipengaruhi oleh jenis zat antara yang dilalui gelombang tersebut sedangkan frekuensi ditentukan oleh sumber getaran.

Menurut rentang frekuensinya, bunyi dibedakan menjadi tiga macam yaitu:

1. Infrasonik dengan frekuensi ≤ 20 Hz.
2. Bunyi yang dapat ditangkap pendengaran normal manusia dengan frekuensi antara 20 hingga 20.000 Hz.
3. Ultrasonik dengan frekuensi ≥ 20.000 Hz.

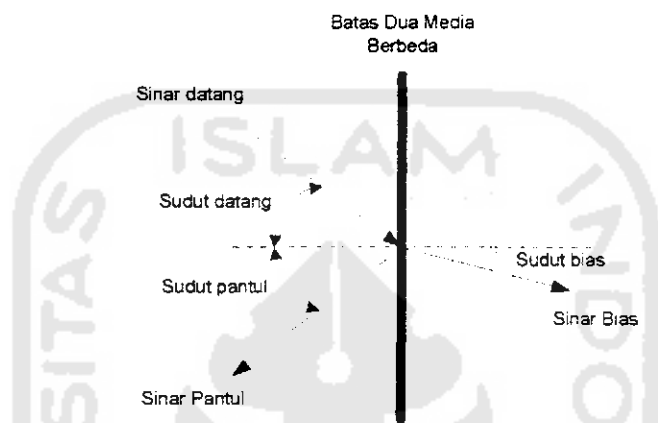
2.1.1 Ultrasonik

Ultrasonik termasuk dalam gelombang bunyi yang memiliki frekuensi di atas 20.000 Hz. Oleh sebab itu sifat – sifat gelombang bunyi pada umumnya juga berlaku padanya. Sifat - sifat penting dari gelombang bunyi tersebut antara lain:

1. Pemantulan.

Pemantulan (*reflection*) akan terjadi bila gelombang mengenai suatu rintangan atau ujung media dimana gelombang tersebut berjalan. Paling tidak sebagian dari gelombang tersebut akan dipantulkan. Pantulan gelombang ditunjukkan

pada gambar 2.1. Sudut yang dibentuk gelombang masuk atau gelombang datang terhadap permukaan pantul adalah sama dengan sudut yang dibentuk oleh gelombang yang dipantulkan. Sehingga terjadi pengertian bahwa sudut datang sama dengan sudut pantul.



Gambar 2.1 Sifat pemantulan dan pembiasan.

2. Interferensi.

Interferensi mengacu pada apa yang terjadi bila dua gelombang melewati daerah ruang yang sama pada waktu bersamaan. Pergeseran resultan pada setiap titik dan waktu adalah penjumlahan dari pergeseran masing – masing secara terpisah.

3. Pembiasan.

Pembiasan (*refraction*) terjadi bila setiap gelombang melewati batas media tertentu. Sebagian energi akan dipantulkan dan sebagian lagi akan dipindahkan atau diserap.

4. Pelenturan.

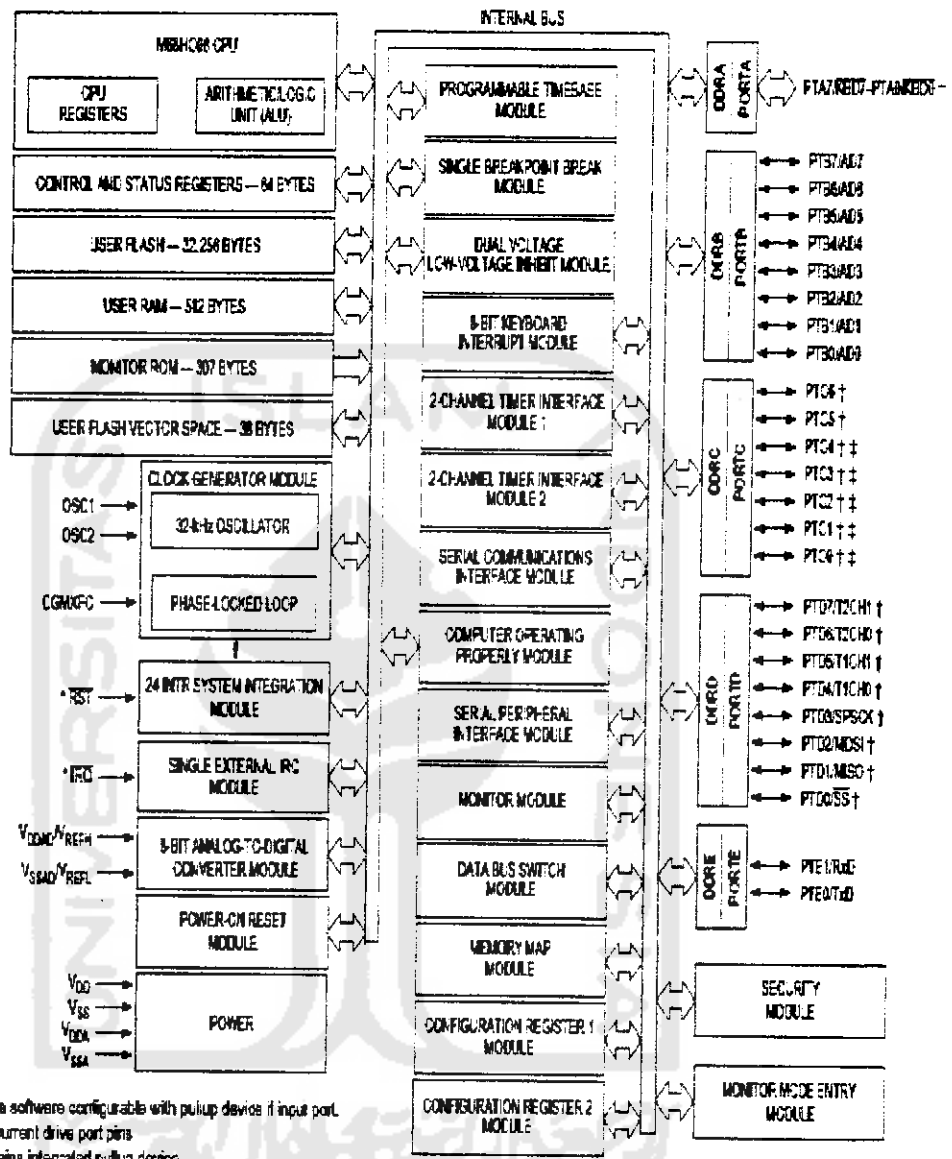
Pelenturan (*diffraction*) merupakan suatu peristiwa dimana gelombang akan mengalami penyebaran ketika sedang berjalan atau ketika menemui rintangan. Jumlah lenturan bergantung pada panjang gelombang dan ukuran rintangan.

2.2 Mikrokontroler MC68HC908GP32

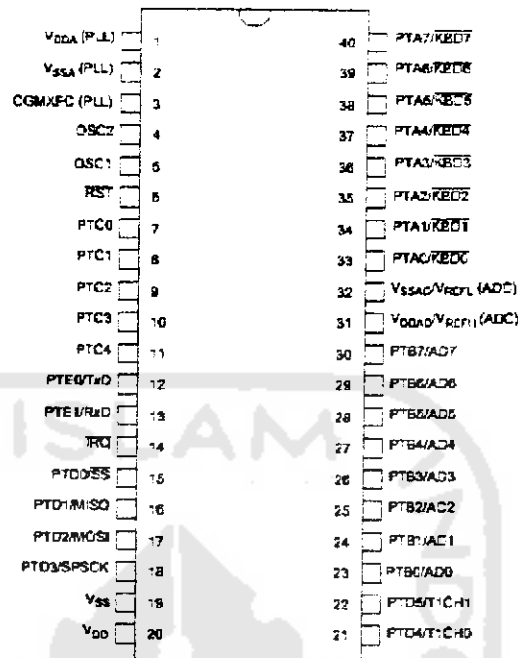
2.2.1 Deskripsi Umum

Mikrokontroler MC68HC908GP32 adalah salah satu mikrokontroler yang diproduksi oleh Motorola. MC68HC908GP32 merupakan mikrokontroler 8 bit dari keluarga MC68HC08. Diagram blok *Microcontroller Unit* (MCU) pada MC68HC908GP32 ditunjukkan pada gambar 2.1.

Tipe *Microcontroller Unit* (MCU) yang digunakan adalah 40 pin *Plastic Dual In Line Package* (PDIP), yang secara garis besar berfungsi sebagai catu daya *Central Processor Unit* (CPU) dan port Input / Output (port I/O) sebagai antarmuka dengan perangkat luar.



Gambar 2.2 Diagram blok mikrokontroler MC68HC908GP32.



Gambar 2.3 MC68HC908GP32 tipe 40 pin.

2.2.2 Fungsi Pin pada MC68HC908GP32

Pin merupakan sarana untuk menghubungkan mikrokontroler dengan perangkat luar. Ada beberapa fungsi pin yang utama yaitu sebagai:

1. Power Supply

V_{DD} dan V_{SS} adalah pin untuk *power supply* sebesar 5 V dan *ground*.

2. Oscillator

OSC1 dan OSC2 terhubung dengan *oscillator internal* dalam chip, bagian ini termasuk dalam *Clock Generation Module (CGMC)* yang bertugas membangkitkan sinyal *clock* kristal.

3. Interrupt

Ada dua macam *interrupt* yang utama yaitu *External Interrupt* dan *Keyboard Interrupt*. Keduanya memiliki fungsi yang sama. Jika pin *interrupt* diberi tegangan 0 V maka sebuah *interrupt* akan menghentikan program dan melayani sub rutin *interrupt* hingga selesai dan kemudian melanjutkan program yang tadi dihentikan.

4. Input dan Output

Pada MC68HC908GP32 tipe 40 pin terdapat 33 pin input / output *bidirectional* dari 5 port paralel yaitu Port A, Port B, Port C, Port D dan Port E. Semua pin I/O dapat diprogram menjadi input atau sebaliknya menjadi output. Pengesetan fungsi input / output dilakukan secara perangkat lunak dengan mengisi register yang berkaitan dengan port yang bersangkutan.

2.2.3 Peta Memori MC68HC908GP32

Central Processor Unit (CPU) 08 yang menjadi basis MCU ini menggunakan alamat 64 Kbytes dari ruang memori. Peta memori diantaranya terdiri dari :

- * 32,256 byte untuk *user FLASH memory*.
- * 512 byte untuk *Random Access Memory* (RAM).
- * 36 byte untuk *user defined vector*.
- * 307 byte untuk *monitor ROM*.

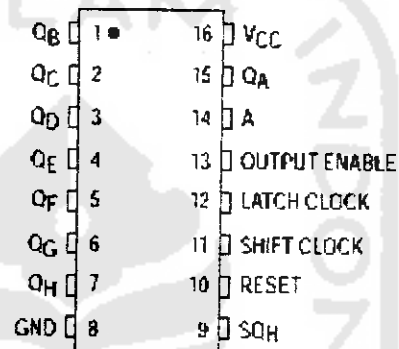
Peta memori mikrokontroler MC68HC908GP32 ditunjukkan pada gambar 2.4.

\$0000	↓	IO Registers 64 Bytes
\$003F		
\$0040	↓	RAM 512 Bytes
\$023F		
\$0240	↓	Unimplemented 32,182 Bytes
\$7FFF		
\$8000	↓	FLASH Memory 32,256 Bytes
\$FDFF		
\$FE00		SIM Break Status Register (SBSR)
\$FE01		SIM Reset Status Register (SRSR)
\$FE02		Reserved (SUBAR)
\$FE03		SIM Break Flag Control Register (SBFCR)
\$FE04		Interrupt Status Register 1 (INT1)
\$FE05		Interrupt Status Register 2 (INT2)
\$FE06		Interrupt Status Register 3 (INT3)
\$FE07		Reserved
\$FE08		FLASH Control Register (FLCR)
\$FE09		Break Address Register High (BRKH)
\$FE0A		Break Address Register Low (BRKL)
\$FE0B		Break Status and Control Register (BRKSCR)
\$FE0C		LVI Status Register (LVISR)
\$FE0D	↓	Unimplemented 9 Bytes
\$FE0F		
\$FE10	↓	Unimplemented 16 Bytes Reserved for Compatibility with Monitor Code for A-Family Parts
\$FE11		
\$FE20	↓	Monitor ROM 337 Bytes
\$1152		
\$1153	↓	Unimplemented 43 Bytes
\$117D		
\$117E		FLASH Block Protect Register (LBPR)
\$117F	↓	Unimplemented 83 Bytes
\$11CB		
\$110C	↓	FLASH Vectors 35 Bytes
\$1111		

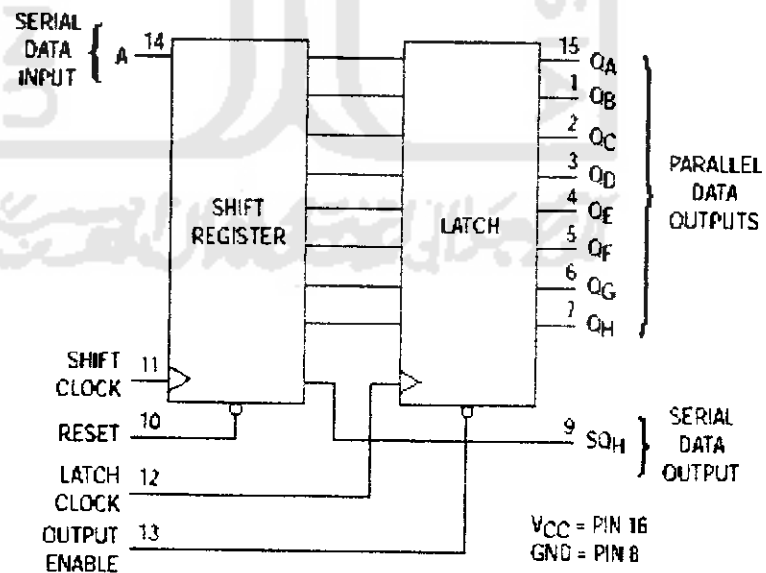
Gambar 2.4 Peta memori MC68HC908GP32.

2.3 MC74HC595A

MC74HC595A adalah IC *shift register*. IC ini berfungsi sebagai pengubah data serial ke dalam bentuk aliran data paralel 8 bit. MC74HC595A berhubungan langsung dengan port data serial pada MCU. Gambar 2.5 adalah posisi pin dari MC74HC595A.



Gambar 2.5 Posisi pin MC74HC595A



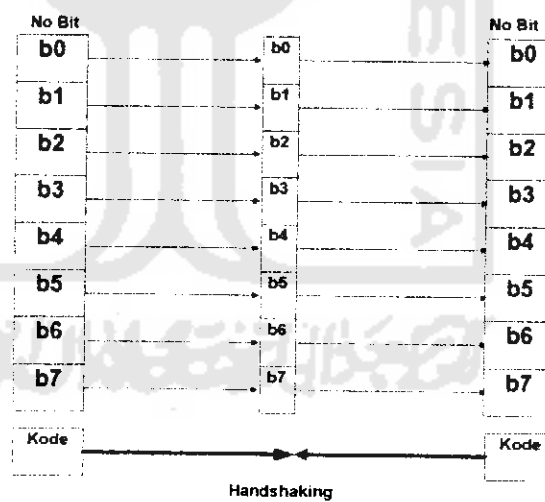
Gambar 2.6 Diagram logika MC74HC595A

2.4 Model Transfer Data

Di dalam komunikasi data dikenal dua sistem transfer data yaitu transfer data secara paralel dan transfer data secara serial. Berikut ini penjelasan dari dua jenis transfer data tersebut.

2.4.1 Transfer Data Secara Paralel

Transfer data paralel dilakukan dengan cara mengirimkan 8 bit data secara serempak menggunakan 8 buah penghantar yang berlainan. Transfer data paralel dilakukan byte demi byte. Delapan bit data tersebut akan sampai tujuan dalam waktu yang bersamaan dan berlangsung dengan kecepatan yang sangat tinggi. Gambar 2.7 menunjukkan proses transfer data secara paralel.

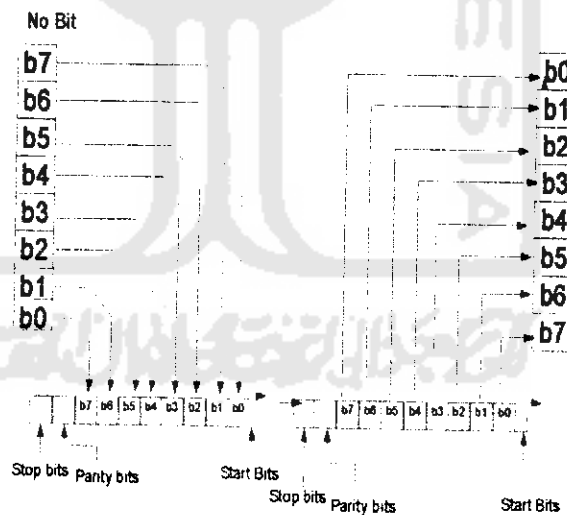


Gambar 2.7 Transfer data secara paralel.

2.4.2 Transfer Data Secara Serial

Pada cara ini bit - bit data dalam bentuk byte dikirimkan secara tunggal berurutan satu persatu. Dari ujung penerimaan, bit - bit tunggal tersebut disusun kembali ke dalam bentuk byte semula. Pengiriman data yang berlangsung secara berurutan bit demi bit pada transfer serial akan mengurangi efektifitas kerja dimana waktu yang dibutuhkan untuk mengirimkan 8 bit data akan lebih lama dibanding pengiriman data secara paralel.

Keuntungan transfer data serial adalah membutuhkan jumlah kabel sedikit sedangkan kerugiannya adalah memiliki kecepatan yang lebih lambat bila dibanding dengan transfer data secara paralel. Proses transfer data secara serial ditunjukkan pada gambar 2.8.



Gambar 2.8 Transfer data secara serial.

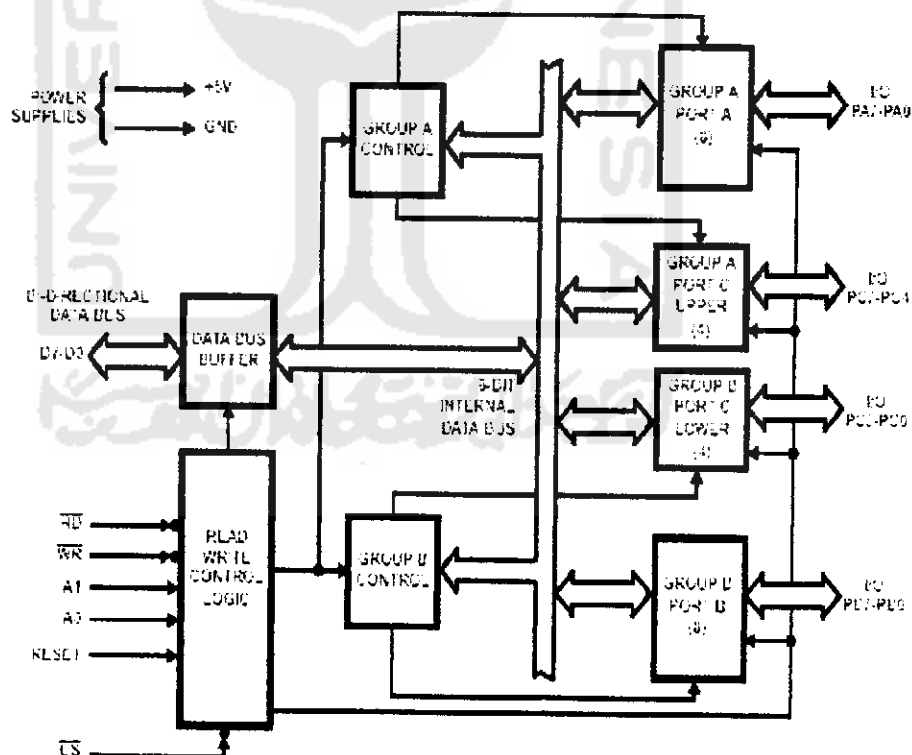
2.5 PPI 8255

2.5.1 Fungsi PPI 8255

PPI 8255 berfungsi sebagai antarmuka. Istilah antarmuka dapat diartikan sebagai titik yang menghubungkan antara dua lingkungan yang berbeda yaitu antara rangkaian dalam komputer dan rangkaian di luar komputer. Antarmuka sering disebut juga sebagai port I/O (port Input / Output).

2.5.2 Arsitektur PPI 8255

PPI 8255 terdiri dari 3 port I/O yaitu Port A, Port B, dan Port C. Masing - masing port terdiri dari 8 bit I/O sehingga total port I/O adalah 24 bit. Diagram blok PPI 8255 ditunjukkan pada gambar 2.9.



Gambar 2.9 Diagram blok PPI 8255

2.5.3 Control Word

PPI 8255 terdiri dari 3 port yaitu Port A, Port B dan Port C yang dapat diprogram sebagai port input maupun port output. Pemilihan konfigurasi port masukan atau keluaran pada PPI 8255 dilakukan dengan cara pengesetan *control word*.

Tabel 2.1 Konfigurasi *Control Word* PPI 8255

Control Word (Hex)	Control Word (Decimal)	Group A		Group B	
		Port A	Port C Upper	Port B	Port C Lower
80	128	OUTPUT	OUTPUT	OUTPUT	OUTPUT
81	129	OUTPUT	OUTPUT	OUTPUT	INPUT
82	130	OUTPUT	OUTPUT	INPUT	OUTPUT
83	131	OUTPUT	OUTPUT	INPUT	INPUT
88	136	OUTPUT	INPUT	OUTPUT	OUTPUT
89	137	OUTPUT	INPUT	OUTPUT	INPUT
8A	138	OUTPUT	INPUT	INPUT	OUTPUT
8B	139	OUTPUT	INPUT	INPUT	INPUT
90	144	INPUT	OUTPUT	OUTPUT	OUTPUT
91	145	INPUT	OUTPUT	OUTPUT	INPUT
92	146	INPUT	OUTPUT	INPUT	OUTPUT
93	147	INPUT	OUTPUT	INPUT	INPUT
98	152	INPUT	INPUT	OUTPUT	OUTPUT
99	153	INPUT	INPUT	OUTPUT	INPUT
9A	154	INPUT	INPUT	INPUT	OUTPUT
9B	155	INPUT	INPUT	INPUT	INPUT

2.6 C++ Builder

C++ Builder adalah perangkat pengembang untuk menghasilkan program aplikasi yang beroperasi dalam lingkungan Windows 9x maupun Windows NT. C++ Builder menggunakan bahasa pemrograman C++ yang diproduksi oleh Borland International, Inc.

Pada saat ini telah banyak perangkat pengembang yang menggunakan bahasa C++. Seperti halnya Borland Delphi, C++ Builder digolongkan dalam kelompok RAD (*Rapid Application Development*) yaitu perangkat pengembang yang mampu dengan mudah dan cepat menghasilkan program aplikasi. C++ Builder dapat membangun suatu program aplikasi dalam tatanan GUI (*Graphical User Interface*) maupun *Console*. GUI adalah karakter program aplikasi yang menggunakan sarana perantara grafis. Seperti kotak – kotak dialog, tombol, menu dan sebagainya. Sedangkan *console* adalah program yang hanya menekankan kemampuan kerja program dan tampilan kurang diperhatikan.

2.6.1 File – File Aplikasi C++ Builder

File aplikasi dicirikan oleh ekstensi .EXE atau .DLL. untuk membangun file jenis ini diperlukan berbagai jenis file seperti ditunjukkan pada tabel berikut ini.

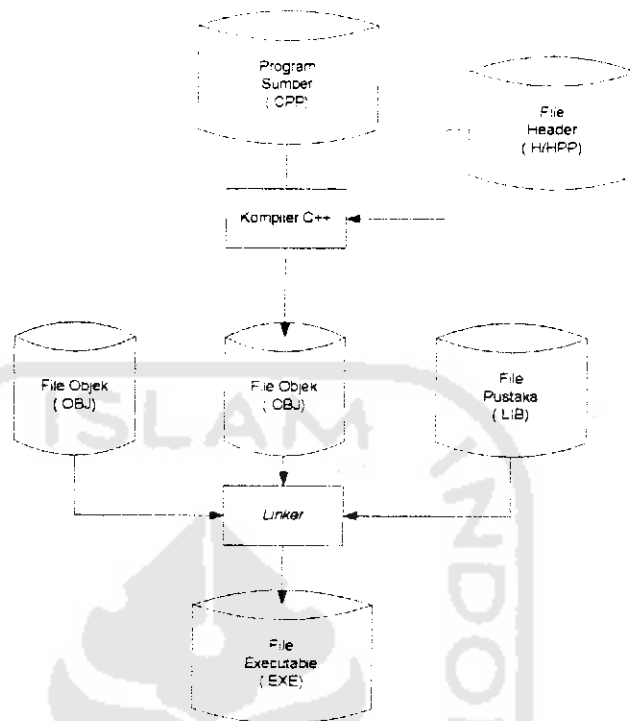
Tabel 2.2 File aplikasi pada C++ Builder

Ekstensi	Keterangan
.CPP	File sumber C++. Pada sebuah proyek, minimal ada dua file dengan ekstensi .CPP ini yaitu file unit yang secara default bernama Unit1.CPP (pada Delphi bernama Unit1.PAS) dan file yang satunya adalah file utama proyek yang secara default bernama Project1.CPP (pada Delphi bernama Project1. DPR).
.DFM	File biner dari form yang digunakan. Informasi yang disimpan dalam file ini adalah hasil pengaturan properti form dan komponen yang ada padanya. Format file ini mirip dengan yang digunakan pada Delphi.
.DSK	File desktop. File ini akan menyimpan informasi tentang komposisi terakhir dekstop C++ Builder pada waktu menutup atau menyimpan suatu proyek. Sehingga apabila nanti membuka proyek tersebut maka susunan dan ukuran masing – masing tampilan akan sama persis dengan komposisi pada waktu penyimpanan.

.H	File header. File ini menampung deklarasi kelas. Pada Delphi, deklarasi kelas dicantumkan pada file unit.
.ILC	Salah satu dari empat file yang dibentuk oleh <i>linker</i> .
.ILD	Salah satu dari empat file yang dibentuk oleh <i>linker</i> .
.ILF	Salah satu dari empat file yang dibentuk oleh <i>linker</i> .
.ILS	Salah satu dari empat file yang dibentuk oleh <i>linker</i> .
.OBJ	File objek biner hasil proses kompilasi. Selanjutnya file ini akan disambung – sambungkan dengan file lainnya sesuai keperluan oleh <i>linker</i> untuk membentuk file dengan ekstensi .EXE.
.MAK	File teks yang berisi kumpulan perintah yang harus dijalankan oleh kompiler pada saat proses kompilasi. Pada Delphi tidak ditemukan file semacam ini namun pada Borland C++ atau pengembang C++ lainnya file ini sering dijumpai.
.RES	File resource yang berisi perlengkapan yang akan dipakai oleh program seperti ikon, cursor, bitmap.
.TDS	File yang dihasilkan oleh Turbo Debugger dalam proses pemeriksaan kesalahan selama kompilasi berlangsung.

2.6.2 Proses Kompilasi

Program C++ selalu ditulis dengan nama ekstensi .CPP. Agar program ini bisa dijalankan (dieksekusi), program harus dikompilasi terlebih dahulu dengan menggunakan kompiler C++. Pada proses kompilasi, program sumber (.CPP) bersama file – file header (.H) akan diterjemahkan oleh kompiler C++ menjadi kode objek (.OBJ). File objek ini berupa file dalam format biner. Selanjutnya file objek ini bersama – sama dengan file objek lain serta file pustaka (.LIB) digabungkan menjadi satu oleh *linker*. Hasil kompilasi berupa file yang bersifat *executable*. File inilah yang bisa dijalankan dari sistem operasi secara langsung.

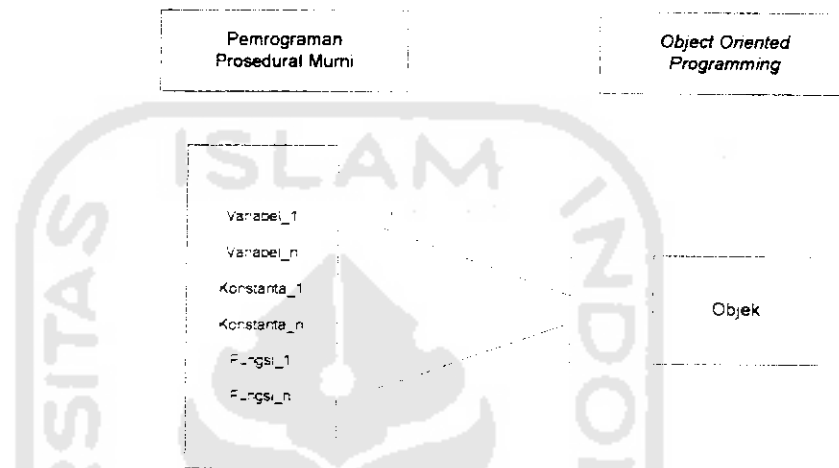


Gambar 2.10 Proses kompilasi C++ Builder

2.6.3 Konsep OOP (*Object Oriented Programming*)

Salah satu kelebihan C++ dibandingkan dengan bahasa pemrograman lainnya adalah kemampuannya dalam pengelolaan program yang lebih diarahkan pada pembentukan objek. Pemrograman semacam ini dinamakan pemrograman berarah objek atau lebih dikenal dengan sebutan OOP (*Object Oriented Programming*). Dengan menerapkan konsep ini program akan lebih mudah untuk dikembangkan. Pemrograman prosedural murni yang tidak menerapkan konsep *object oriented* banyak menitikberatkan ke arah pembentukan fungsi – fungsi sehingga di dalam program akan terdapat banyak sekali fungsi dan variabel yang tidak tertata rapi. Dari kelemahan tersebut, dibentuklah bahasa yang menerapkan

OOP untuk menyederhanakan fungsi – fungsi dan variabel – variabel ke dalam bentuk objek. Berikut ini gambaran perbedaan yang terdapat pada pemrograman prosedural murni dengan OOP.



Gambar 2.11 Konsep OOP

Dari gambaran tersebut sebuah objek dapat didefinisikan sebagai sesuatu yang mempunyai data (berupa variabel atau konstanta) dan perilaku (berupa fungsi). Jika bekerja di dalam lingkungan C++ Builder, data – data milik objek tersebut disebut *properti* dan perilaku yang dimilikinya disebut *method*. Adapun kejadian – kejadian yang dapat menimpa objek tersebut akan dinamakan *event*. Secara garis besar yang menjadi ciri dari OOP adalah terdapatnya proses abstraksi (*abstraction*), pengkapsulan (*encapsulation*), penurunan sifat (*inheritance*) dan polimorfisme (*polymorphism*) pada objek – objek yang dibentuk.